

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA  
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh a realizace SW aplikace ve firmě

Design and Implementation of SW Application in a Company

Student: Miroslav Hon

Vedoucí diplomové práce: Ing. Alena Juráková, Ph.D.

Ostrava 2009

Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.

V Ostravě dne 28.4.2009

.....

Miroslav Hon

## **Poděkování**

Rád bych poděkoval vedoucí mé diplomové práce Ing. Aleně Jurákové, Ph.D. za poskytnutí velmi cenných rad a odborné vedení této práce.

Rovněž velké poděkování patří pracovníkům oddělení Akvizic centra zákaznických služeb společnosti GE Money v Ostravě za spolupráci při tvorbě návrhu a prototypu aplikace, a samotné společnosti GE Money za umožnění vypracování diplomové práce.

# Obsah

Úvod.....	1
1 Teoretická východiska.....	2
1.1 Specifikace požadavků.....	2
1.1.1 Způsoby zpracování požadavků.....	2
1.2 UML.....	4
1.2.1 Struktura jazyka UML.....	4
1.2.2 Diagramy UML.....	6
1.2.2.1 Diagram případů užití (Use Case diagram).....	7
1.2.2.2 Diagram tříd (Class diagram).....	9
1.2.2.3 Objektový diagram (Object diagram).....	11
1.2.2.4 Diagram aktivit (Activity diagram).....	12
1.2.2.5 Sekvenční diagram (Sequence diagram).....	14
1.2.2.6 Komunikační diagram (Communication diagram).....	15
1.2.2.7 Ostatní diagramy.....	16
1.3 Důvody výběru popsaných nástrojů.....	17
2 Analýza současného stavu.....	18
2.1 O společnosti.....	18
2.2 Objekt řešení.....	19
2.3 Současný stav.....	19
3 Návrh a realizace řešení.....	21
3.1 Optimalizace procesu a algoritmus výpočtu.....	22
3.2 Specifikace požadavků.....	23
3.2.1 Seznam požadavků.....	23
3.2.2 Popis požadavků.....	24
3.3 Modelování případu užití.....	27
3.3.1 Specifikace účastníků.....	28
3.3.2 Matice RTM.....	29
3.3.3 Specifikace případu užití.....	30
3.4 Analýza případu užití.....	31
3.4.1 Model tříd.....	31
3.4.1.1 Analyzovaný text.....	31
3.5 Realizace případu užití – chování systému.....	37
3.5.1 Sekvenční diagramy.....	37

3.5.2	Diagramy aktivit.....	40
3.6	Vytvoření prototypu .....	43
3.7	Návrh budoucí softwarové realizace .....	43
3.8	Předběžný časový plán projektu.....	44
4	Zhodnocení efektivnosti navrhovaného řešení.....	47
4.1	Očekávané náklady .....	47
4.1.1	Jednorázové náklady .....	47
4.1.2	Provozní náklady.....	48
4.2	Přínosy.....	48
4.3	Zhodnocení efektivnosti.....	50
	Závěr .....	51
	Seznam použité literatury .....	53
	Seznam zkratk .....	54
	Seznam tabulek .....	55
	Seznam obrázků.....	55
	Seznam Příloh .....	56
	Prohlášení o využití výsledků diplomové práce .....	57
	Přílohy.....	58

# Úvod

Pro dnešní dobu je charakteristické, že do každodenního života zasahují informace, které dennodenně získáváme a musíme je zpracovat. Nezbytnou součástí tohoto počínání jsou informační technologie napomáhající zefektivňování lidské práce a ve své podstatě také ulehčující lidské žití. V současnosti není problém získat velké množství informací, ale problém nastává u nalezení správného přístupu ke zpracování dat, selekce množiny podstatných dat a hlavně při vytěžování užitečných informací vhodných pro naše účely.

V tvrdém konkurenčním prostředí dnešního podnikání jsou informační technologie silnou strategickou zbraní. Nejenže se při jejich využití dostáváme k vyšší efektivitě práce a snižování nákladů, ale napomáhají také k odhalování nových souvislostí a k vytváření predikcí budoucího vývoje společnosti popřípadě trhu.

Každá pravidelně se opakující činnost bývá většinou algoritmizovatelná, a tudíž převeditelná do podoby softwaru ulehčující tuto činnost. V konkrétní podobě se dá tento nástroj aplikovat na prémiové modely společností, které se pravidelně používají s měsíční periodicitou a mají za úkol ohodnotit práci zaměstnanců. Každý takový model musí obsahovat množinu pravidel zakomponovaných do algoritmu výpočtu, zabezpečující spravedlivé rozdělení prémie pro všechny zaměstnance pracující současně na různorodě měřených činnostech. Zajištění funkčnosti prémiového modelu často podporuje software přizpůsobený této problematice.

Cílem mé práce je navrhnout aplikaci spravující prémiový model oddělení Akvizic centra zákaznických služeb společnosti GE Money v České republice. Aplikace musí splňovat všechny podmínky pro zajištění celkového prémiového modelu, což obsahuje evidenci potřebných dat, flexibilitu v nastavení pravidel výpočtu, provádění výpočtů a přerozdělení prémie, a hlavně automatizaci některých činností za účelem zefektivnění práce zaměstnanců. Aplikace musí být napojena na firemní databáze a systémy z důvodu účinného získávání a předávání informací. Jelikož proběhla optimalizace procesu a tvorba nového algoritmu výpočtu pro prémiový systém, klade vedení důraz na vytvoření prototypu aplikace, ve kterém by bylo možné proces i algoritmus nasimulovat. Aplikace musí rovněž splňovat další požadavek vedení oddělení, kterým je zajištění návratnosti tvorby aplikace do dvou let.

# 1 Teoretická východiska

## 1.1 Specifikace požadavků

Z hlediska uživatele závisí funkčnost informačního systému nebo aplikace na kvalitní interakci vývojového týmu a uživatelů. Hlavní roli při vývoji softwaru hraje správné pochopení potřeb uživatelů. Souborem aktivit zabývajících se procesem specifikace potřeb uživatelů se nazývá „requirements Engineering“ (inženýrství požadavků). Pod pojmem požadavek rozumíme určitou funkci nebo schopnost, která splňuje potřebu uživatele a měla by být implementována v realizovaném systému [8]. Požadavky dělíme na dva základní typy:

- **funkční** – požadavky týkající se funkčností nebo služeb, které má daný systém splňovat.
- **nefunkční** – netýkají se funkcí systému, ale vlastností a omezujících podmínek (např. typ interakce systému a uživatele, vzhled, rychlost odezvy, velikost na disku atd.).

Formulace požadavků odpovídá nastaveným standardům využívaných v daném prostředí. Nejobecněji mohou být popsány pomocí přirozeného jazyka, ve složitějších formách pak pomocí strukturovaného jazyka, grafickém či matematickém vyjádření.

### 1.1.1 Způsoby zpracování požadavků

Jedná se o složitý podproces vývoje softwarových systémů, během kterého analytik s uživatelem shromažďují všechny dostupné informace k vytvoření modelu požadavků dané problémové oblasti. Mezi hlavní způsoby získávání požadavků patří:

- **přímá komunikace s uživatelem** – při tomto způsobu dochází k přímé interakci uživatele a analytika. Kvalita požadavků získaných tímto způsobem závisí na zkušenostech analytika a na vhodném výběru uživatelů, kteří by měli mít jasnou představu o budoucí podobě systému. Analytik má možnost využít mnoha technik, z nichž nejčastější jsou:
  - open-end rozhovor – ve formálně uvolněné atmosféře uživatel hovoří o své práci. Tato technika slouží pro získání celkového obrazu o problémové oblasti a nalezení hlavních požadavků na systém.
  - strukturovaný rozhovor – analytikem řízená diskuze, při které uživatel rovněž hovoří o své práci, ale výklad je doplňován připravenými otázkami analytika.

Technika vede k vytipování hlavních problémů a doplnění informačních mezer analyzované oblasti.

- kolektivní přijímání rozhodnutí – moderní techniky brainstormingu a workshopů umožňují odhalení hlavních, ale i složitě identifikovatelných požadavků na systém. Hrozí zde však nebezpečí psychologických zábran ve skupinovém vyjadřování některých účastníků.
- dotazníky – neosobní komunikace, šetří náklady a čas, poskytují možnost oslovení více respondentů a tím pádem získání více informací.
- **pozorování** – analytik často využívá tuto metodu pro získání celkového obrazu chování systému organizace. Pozorování se dotýká jak práce jednotlivých uživatelů, tak hlavních procesů organizace. Častokrát zkušený analytik dokáže při pozorování identifikovat hlavní problémy a nedostatky ve firemních procesech, které by mohly být v budoucím systému eliminovány.
- **analýza cílů** – při specifikaci požadavků musíme skloubit potřeby uživatelů ze všech úrovní řízení, a rovněž bychom neměli zapomínat na podnikatelské cíle a firemní strategii, kterou by měl odrážet vyvíjený systém. Cíle bývají jasně popsány jako určitý budoucí stav, ale méně jsou konkretizovány jednotlivé kroky a akce k dosažení těchto cílů.
- **tvorba scénářů** – scénářem rozumíme určitý popis pracovního postupu modelující chování systému, které probíhá v interakci s uživatelem. Scénář umožňuje poměrně jednoduchý způsob získání zkušeností od uživatele, kdy tento aktér snáze předá své zkušenosti a znalosti analytikovi, a to v průběhu aktivní tvorby scénářů popisujících jeho každodenní pracovní postupy.
- **analýza formulářů** – většina dnešních společností využívá formuláře jako hlavní komunikační kanál. Formulář je formalizovaný dokument obsahující množinu proměnných, které jsou vybírány za účelem sběru dat dané problémové oblasti. Formuláře obsahují dostupné a důležité informace a často se stávají základem pro tvorbu funkčního modelu systému.

Požadavkům lze rovněž přidělit prioritu, podle které k nim bude dále přistupováno. Rozsah prioritizace se dá převzít z již nadefinovaných metodik nebo je možné určit vlastní stupnici priority.



## 1.2 UML

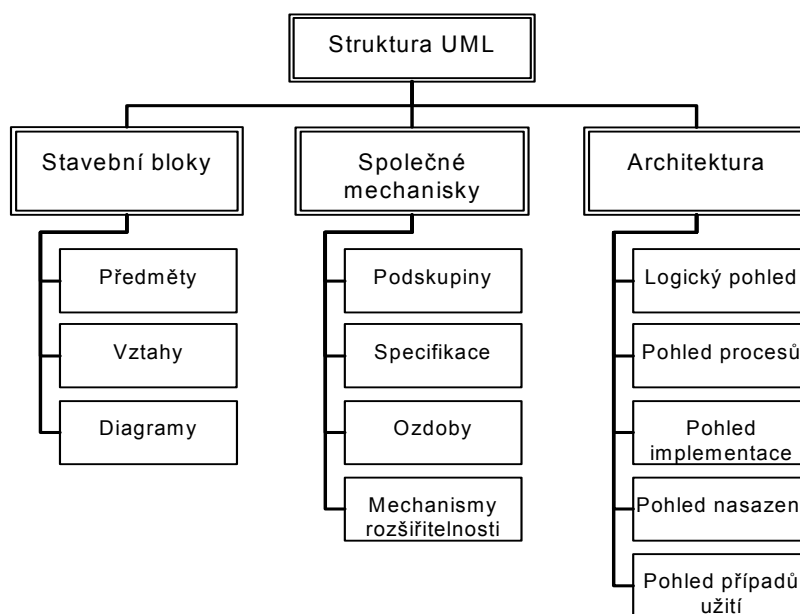
Jazyk UML (Unified Modeling Language) je univerzální jazyk pro vizuální objektově orientované modelování systémů [1]. Tento nástroj byl vytvořen předními kapacitami v oboru vývoje informačních technologií (Booch, Rumbaugh, Jacobson). Jazyk byl navržen, aby umožnil propojení nejlepších existujících postupů a modelovacích technik z oblasti softwarového inženýrství. Jazyk UML nenabízí žádný druh metodiky modelování, ale pouze vizuální syntaxi využitelnou pro sestavování jednotlivých modelů. Je explicitně navržen s ohledem na možnost implementace nástrojů CASE (Computer-Aided Software Engineering), bez kterých si vytváření komplexnějších systémů umíme jen těžko představit [1].

UML obsahuje bohatou škálu sémantických prvků a syntaxí, které umožňují snadný návrh a vizualizaci aplikace jakéhokoliv druhu. Navíc poskytuje zabudované mechanismy rozšíření pro případy, kdy do modelu zavádíme nové prvky nebo nové elementy jazyka. UML spadá mezi skutečně univerzální nástroje a dokazuje to možnost využití některých jeho prvků pro vizuální modelování v jiných oblastech, jako je procesní řízení.

Jakožto průmyslový standard prochází jazyk určitým vývojem a postupem času se dále zdokonaluje a rozšiřuje. V současnosti se pracuje s verzí UML 2.2.

### 1.2.1 Struktura jazyka UML

Základní strukturu jazyka UML tvoří tři hlavní bloky, které znázorňuje obrázek 1.1.



Obr. 1.1 Struktura UML.

- **Stavební bloky** – se v jazyce UML skládají z těchto hlavních prvků:
  - *Předměty* – představují samotné prvky modelu (třídy, rozhraní, uzly, komponenty, interakce atd.).
  - *Vztahy* – vytvářejí spojení mezi předměty. Deklarují vzájemnou souvislost dvou a více předmětů.
  - *Diagramy* – představují pohledy na modely UML, které znázorňují určitý příběh o softwarovém systému a jsou způsobem vizualizace, *co* systém bude dělat a *jak* to bude dělat [2].
  
- **Společné mechanismy** – popisují způsoby dosažení specifických cílů jazyka UML.
  - *Podskupiny* – diverzifikují různé způsoby vidění světa. V UML se dělí na dvě hlavní podskupiny: skupina klasifikátorů a instancí a skupina rozhraní a implementací.
  - *Specifikace* – textové vyjádření sémantiky jednotlivých elementů diagramů. Množina specifikací tvoří jádro modelu a utváří soudržný sémantický podklad.
  - *Ozdoby (nebo také Ornamenty)* – slouží ke zdůraznění a zvýraznění důležitých detailů prvků modelu (např. jednoduché vyjádření viditelnosti: „+“ veřejný, „–“ soukromý atd.).
  - *Mechanismy rozšiřitelnosti* – tvůrci jazyka byli obezřetní a uvědomovali si nemožnost vytvoření stoprocentně univerzálního řešení, a proto zavedli jednoduché mechanismy rozšiřitelnosti. Patří mezi ně tyto mechanismy: omezení, stereotypy, označené hodnoty.
  
- **Architektura** – představuje organizační strukturu systému, včetně jeho rozkladu na součásti, jeho propojitelnosti, interakce, mechanismů a směrných zásad [5]. Nejznámějším pohledem na architekturu je „4+1“ popsany níže:
  - Logický pohled – slovník problémové oblasti zobrazující množinu tříd a objektů včetně zachycení jejich chování.
  - Pohled procesů – navazuje na logický pohled se zaměřením na procesní orientaci modelu.
  - Pohled implementace – ukazuje závislosti jednotlivých komponent včetně konfigurace množin z těchto komponent.

- Pohled nasazení – modeluje fyzické nasazení komponent na množinu fyzických výpočetních uzlů (počítačů a periferních zařízení) [1].
- Pohled případu užití – základ, od kterého jsou odvozeny ostatní pohledy. Souhrn případů užití zachycuje hlavní požadavky na navrhovaný systém.

### 1.2.2 Diagramy UML

Jedná se o hlavní stavební bloky jazyka UML, jenž lze rozdělit dle dvou aspektů, a to zda popisují strukturu (popisy předmětů, strukturní asociace atd.) nebo chování systému (znázornění požadovaného chování systému). Rozdělení diagramů viz tabulka 1.1.

Struktura	Chování	
Diagram tříd	Diagram případů užití	
Objektový diagram	Diagram aktivit	
Diagram komponent	Stavový diagram	
Diagram nasazení	Diagramy interakce	- Sekvenční diagram
Diagram složené struktury		- Komunikační diagram
Diagram balíčků		- Diagram časování
		- Stručný diagram interakce

Tab. 1.1 Rozdělení diagramů.

Diagramem rozumíme souhrn grafických prvků poskládaných podle zřejmých pravidel do podoby požadovaného schématu. Diagram není modelem. Diagram znázorňuje určitý průřez popisující náš pohled na daný model/systém. Modelem rozumíme souhrn všech předmětů a vztahů, jejichž jednoduché znázornění je téměř nemožné. Navíc z diagramů lze odstranit předměty a vztahy, ale v modelu tyto prvky mohou stále existovat.

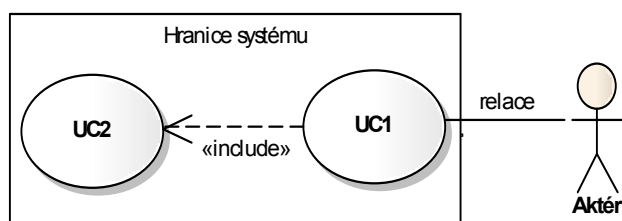
V UML není přesně stanovené pořadí, v němž by se měly diagramy vytvářet. Přesto se obvykle začíná diagramem případu užití, který definuje rozsah platnosti navrhovaného systému. Na něj navazuje identifikace tříd, rozpracování chování systému v podobě diagramu aktivit a interakce. Ve skutečnosti často pracujeme souběžně s několika různými diagramy zároveň a při jejich tvorbě jsou postupně odhalovány detaily navrhovaného systému [1].

### 1.2.2.1 Diagram případů užití (Use Case diagram)

V UML se tento diagram využívá jako jiný způsob zachycení a dokumentování požadavků modelovaného systému. Specifikace požadavků, popsaná výše, často poskytuje podklad pro vytváření Use Case diagramu (dále jen UC). Tento diagram slouží pro grafické zobrazení fungujícího systému složeného z jednotlivých případů užití (neboli funkcionalit). Grafický model zachycuje sice celkovou podobu a strukturu systému složeného z UC, ale detaily popisující fungování a algoritmus jednotlivých UC se definují v textové podobě specifikace UC. I když se modelování UC může zdát jako jednoduchý proces, měli bychom mu věnovat patřičnou pozornost, jelikož pouze navržené případy užití budou dále zapracovávány do nového systému. Častokrát však dochází k nevyhnutelným změnám ve specifikacích UC z důvodu odhalování nových skutečností v rámci postupných fází modelování.

Model případu užití se skládá ze čtyř hlavních komponent (viz obr. 1.2):

- Hranice systému (System boundary) – hranice zřetelně označující rozhraní mezi prvky modelovaného systému a okolním světem (subjekty spolupracující se systémem).
- Aktéři (Actors) – aktéry lze definovat jako externí entity (předměty a role osob) spolupracující s modelovaným systémem.
- Případy Užití (UC- Use Cases) – případy užití popisují funkce, které systém poskytuje k užítku jednoho nebo více aktérů [5].
- Relace (Relationship) – vztahy mezi komponenty UC diagramů. Komunikační relace spojuje aktéry a případy užití. Relace typu „include“ se využívá v případě, kdy je jeden UC využíván jinými UC. Směr této relace ukazuje na využívaný UC. Relace typu „extend“ využíváme pro spojení UC, kde jeden UC rozšiřuje funkcionality jiného UC. Relace „generalization“ má obdobný smysl jako v modulu tříd, a to zobecnění určité komponenty.



Obr. 1.2 Komponenty UC diagramů.

Nejčastěji začínáme modelování případů užití nalezením hranic systému umožňujících dospět k definici rozsahu problémové oblasti. Následné akce bývají prováděny souběžně a opakovaně. Mezi následné činnosti patří identifikace aktérů, nalezení případů užití a definování relací. Pro případy užití vypracováváme specifikaci případů užití, která poskytuje sémantický popis UC. UML nenabízí pro specifikaci žádný standard, ale pouze doporučení a šablony. Obecně bychom se při tvorbě specifikace měli držet jednoduché formy. Při specifikaci bychom neměli zapomínat na tyto prvky:

- Jedinečný identifikátor a název UC.
- Stručný popis zachycující podstatu řešeného problému pomocí UC.
- Aktéři hrající nějakou roli v UC.
- Vstupní podmínky týkající se stavu systému a omezující spuštění UC.
- Hlavní scénář popisující sled následných kroků UC.
- Alternativní scénář odkazující na seznam alternativních scénářů vůči hlavnímu scénáři.
- Výstupní podmínky omezující výstupy a konec případu užití.

Ve specifikacích se mohou vyskytovat další části rozšiřující popis daného UC avšak záleží na potřebách zúčastněných stran modelování. Tyto strany by si měly stanovit jednotný formát specifikace používaný v celém projektu.

Jelikož se specifikace uživatelských požadavků a případy užití dívají na problém odlišným pohledem, musíme nastavit aspekt sledování požadavků a UC, abychom tyto dvě oblasti propojili a nevznikaly v žádné z nich odlišnosti. Pro sledování lze využít CASE nástrojů. Pokud ale nemáme žádný nástroj, lze tento problém jednoduše vyřešit pomocí matice RTM (Requirements Traceability Matrix – matice sledovanosti požadavků). Tabulka 1.2 znázorňuje jednoduché vyjádření sledování požadavků, kdy ve svislé ose jsou vyznačeny požadavky a ve vodorovné případy užití. Křížkováním zaznamenáme splnění požadavků jednotlivými UC.

Případy užití					
Požadavky		UC1	UC2	UC3	UC4
	R <sub>1</sub>				
	R <sub>2</sub>				
	R <sub>3</sub>				
	R <sub>4</sub>				

Tab. 1.2 Matice RTM.

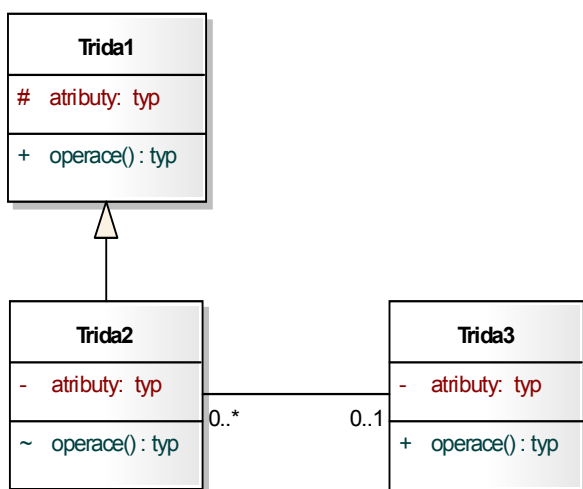
### 1.2.2.2 Diagram tříd (Class diagram)

Diagram tříd, jako statický pohled na systém, zobrazuje množinu tříd včetně statických vztahů mezi nimi. V UML lze dále diagramy tříd rozdělit dle fáze, ve které se využívají (např. analytické třídy, návrhové třídy). Pod pojmem „třída“ rozumíme přesně specifikovaný předpis definující charakteristické vlastnosti (název, atributy, metody atd.) určité množiny objektů. Objekt vznikne vytvořením instance vybrané třídy. Objekt je každý prvek reálného světa, jenž v objektově orientovaném programování (dále OOP) modelujeme do abstraktní podoby (vlastnosti objektu včetně chování), čímž napodobujeme realitu.

Objektový přístup se řídí jasně danými paradigmaty, která jsou:

- Abstrakce – princip identifikace a oddělení podstatných vlastností objektů reálného světa od těch nepodstatných.
- Zapouzdření – skrytí vnitřních vlastností a chování daného objektu před ostatními objekty.
- Dědičnost – princip odpovídající reálnému světu, kdy jeden objekt může dědit množinu vlastností jiných objektů.
- Polymorfismus – neboli mnohotvarost, znamená schopnost jednoho algoritmu nabývat různých stavů a výstupů.

Třidu v jazyce UML znázorňuje obdélník rozdělený na tři části (viz obr. 1.3). V záhlaví se vyskytuje jméno třídy, prostřední část obsahuje vlastnosti (atributy definující statickou strukturu) a dolní část operace (znázorňující chování).



Obr. 1.3 Notace tříd.





Atributy a operace mají definován datový typ (u operace návratový typ) stanovující množinu přípustných hodnot, a viditelnost v podobě ornamentu (definovány v tabulce 1.3). Dále se v notaci operace nachází seznam parametrů ovlivňující algoritmus operace.

Ornament	Typ viditelnosti	Popis
+	Public	S atributem nebo operací mohou pracovat všechny prvky.
-	Private	Umožněno použití pouze uvnitř dané třídy.
#	Protected	Umožněno použití pouze uvnitř třídy a potomkům dané třídy.
~	Package	Umožněn přístup v rámci téhož balíčku a podbalíčků.

Tab. 1.3 Definice ornamentů viditelnosti.

Diagram tříd se skládá z množiny tříd, které spojuje množina čar určující vzájemné vztahy tříd. Tyto vztahy se nazývají asociace a rovněž mají své vlastnosti a jsou různých typů. Mezi nepovinné vlastnosti asociace spadá ***název asociace*** (sloveso jednoznačně identifikující a popisující asociaci), ***název role*** (popis role dané třídy), ***násobnost*** (označení množiny objektů všech tříd relace, které lze v jednom okamžiku zahrnout do vztahu) a ***průchodnost*** (šipka na příslušném konci relace určující směr průchodnosti informací).

Asociace nemají jen své vlastnosti, ale také se mohou řadit do různých typů, které popisuje tabulka 1.4.

Označení	Název typu	Popis
	Běžná asociace	Běžná asociace s naznačeným směrem průchodnosti. Asociace bez šipky znamená obousměrnou průchodnost.
	Generalizace	Znázorňuje vztah obecné třídy s jejími potomky, které využívají principu dědičnosti a tím přebírají vlastnosti a chování obecné třídy. Specializaci lze vyjádřit obráceným pohledem na generalizaci, kdy vytvořením podtřídy a rozšířením obecné třídy specializujeme vlastnosti nové třídy.
	Agregace	Pokud se jeden objekt stává součástí druhého, jedná se o agregaci. Hlavní objekt využívá podřízený objekt a je zodpovědný za jeho vznik a destrukci.
	Kompozice	Speciální typ silnější formy agregace, kdy nadřízený objekt nedává smysl bez podřízeného objektu.

Tab. 1.4 Typy asociací.

Postup hledání dokonalých analytických tříd není nikterak jednoduchý proces, jelikož se v průběhu navrhování systému objevují nové skutečnosti a faktory ovlivňující konečnou podobu a chování systému. Rychlost nalezení optimální podoby tříd závisí opět na zkušenostech a znalostech analytika. Existuje mnoho způsobů jak dospět k analytickým třídám a UML definuje následující:

- **Analýza podstatných jmen a sloves** – způsob spočívá ve shromažďování podstatných informací o dané problémové oblasti a následné analýze. Kandidátní třídy a atributy hledáme nalezením podstatných jmen a spojení podstatných jmen. Kandidátní odpovědnosti a operace analyzujeme za pomoci vyhledání sloves a slovesných frází. Takto nalezené kandidátní prvky dále analyzujeme a vytváříme kandidátní třídy, kterým dále přiřazujeme atributy a odpovědnosti. Při tvorbě prvních asociací vycházíme jak z výše popsané analýzy, tak z dalších zdrojů, jakým může být analýza případu užití.
- **Metoda štítků CRC** – jedná se o kolektivní metodu nalézání analytických tříd, kdy zábavnou formou spontánní diskuze hledáme nápady a inspiraci. Podstatné předměty problémové oblasti se zaznamenávají na lístečky rozdělené na tři části – oddíl s názvem třídy, oddíl se seznamem odpovědností a oddíl se seznamem spolupracovníků. Takto připravená data se dále analyzují a zpracovávají do analytických tříd a asociací.

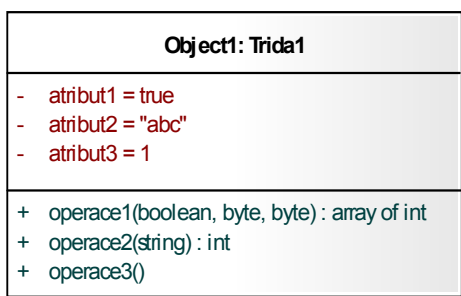
Mnoho autorů doporučuje společné využívání obou technik, včetně analýzy případů užití a specifikace uživatelských požadavků.

### 1.2.2.3 Objektový diagram (Object diagram)

Objektový diagram, *vycházející z diagramu tříd*, znázorňuje objekty a jejich vztah v určitém čase. Podobá se snímku části běžícího objektově orientovaného systému v konkrétním okamžiku, na němž jsou zachyceny aktuální objekty a vazby mezi nimi [1]. Objekty v objektovém diagramu jsou instancí tříd z diagramu tříd, spojení (Link) mezi objekty pak instancí asociací.

Notace objektů se odráží z diagramů tříd a objekt se tedy skládá ze tří oddělených částí (viz. obr. 1.4).





Obr. 1.4 Notace tříd.

Při úplné specifikaci objektu nalezneme v první části název objektu, včetně názvu třídy, z které byla vytvořena instance objektu. Druhá část reprezentuje vlastnosti objektu, a tudíž obsahuje všechny atributy, včetně inicializačních hodnot. Třetí část zastupují operace zprostředkovávající chování objektu.

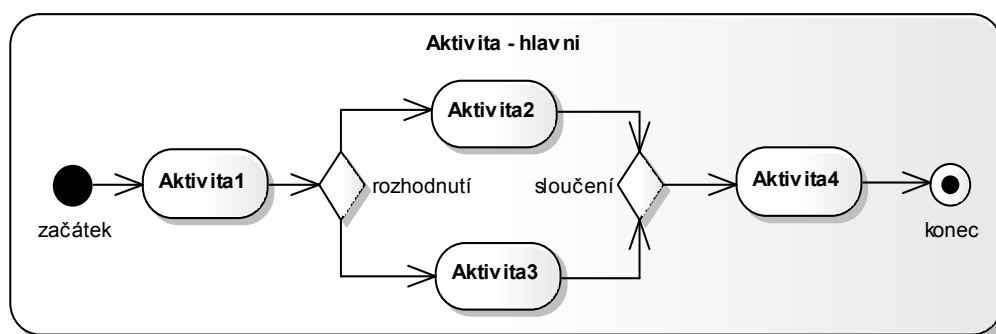
Objekty mezi sebou komunikují zasíláním zpráv, což v objektovém diagramu reprezentují spojení. Zasílání zpráv podmiňuje existence objektu a viditelnost operací, popřípadě atributů objektu.

Diagram tříd rovněž umožňuje znázornění objektů, a proto mnozí analytici při řešení méně složitých problémů vynechávají objektový diagram a objektové prvky značí přímo v diagramu tříd.

#### 1.2.2.4 Diagram aktivit (Activity diagram)

Též definovány jako objektově orientované vývojové diagramy jsou použitelné k modelování všech druhů procesů. Diagramy aktivit lze znázornit a zachytit chování libovolného elementu (např. případu užití, třídy, rozhraní, komponenty, spolupráce).

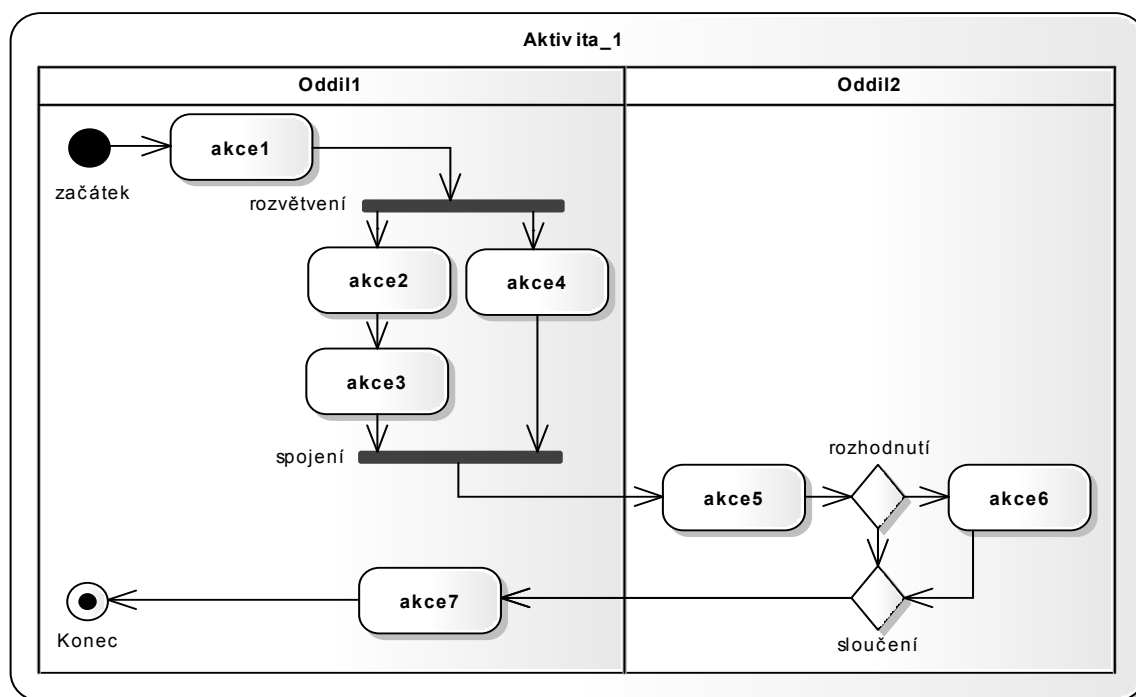
Robustnost UML potvrzuje schopnost zaznamenání paralelních procesů a jejich následnou synchronizaci pomocí aktivity diagramů. Tyto diagramy je možno využít během všech fází modelování (analýza, návrh atd.).



Obr. 1.5 Diagram aktivit.

Hlavním prvkem modelu je aktivita označována obdélníkem se zaoblenými rohy. V nejširším pojetí modelu zakreslujeme hlavní aktivity znázorňující určité procesy nebo chování systému (viz obr. 1.5). Dále rozpracováváme jednotlivé aktivity až do detailu, kdy aktivita tvoří hranici a uvnitř aktivity se vyskytují již nerozložitelné akce (viz obr. 1.6). Opět tento postup nemusí být standardem a mnozí analytici přistupují až k poslední fázi modelování aktivity s konkrétními akcemi (např. realizace konkrétního případu užití).

Každá aktivita má svůj *začátek* a *konec* ( kurzívou označený text poukazuje na prvky uvedené v obrázku 1.6). Již nedělitelnou činnost reprezentuje *akce* (nebo také uzel). Přechod z jedné aktivity do druhé iniciovaný ukončením předchozí akce znázorňuje čára s šipkou. Prvek *rozhodnutí* umožňuje podmíněné alternativní větvení. Pomocí stejného elementu označujeme *sloučení* po alternativním větvení. Pro realizaci paralelního větvení slouží prvek *rozvětvení*. Po tomto paralelním rozvětvení musíme v určitém bodě průběh aktivity synchronizovat, a to za pomoci elementu *spojení*. Pro vyjádření odpovědností za realizaci všech akcí rozdělujeme a seskupujeme související akce do *oddílů*, často též označovaných jako zóny odpovědností (swimlanes). Ty mohou reprezentovat třídy, komponenty, případy užití, aktéry atd.



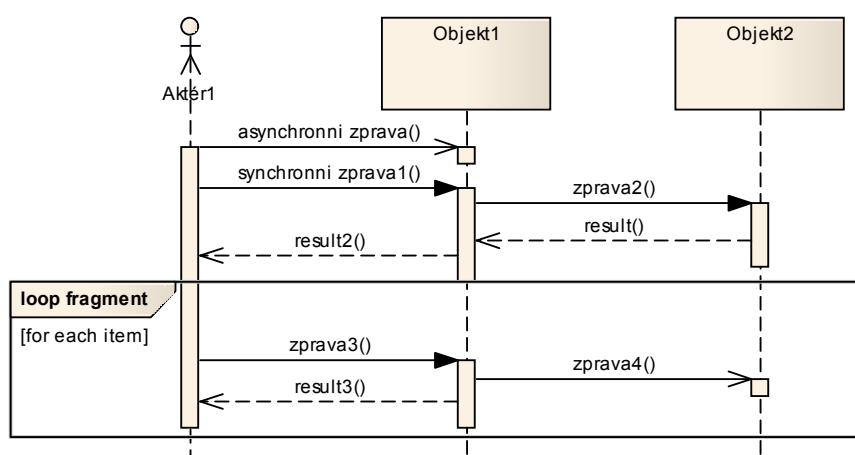
Obr. 1.6 Podrobnější diagram aktivit.

Diagramy aktivit připouští zobrazení dalších elementů, které se nevyskytují na obrázku 1.6 (zasílání signálů, objekty, časová událost).

### 1.2.2.5 Sekvenční diagram (Sequence diagram)

Společně s komunikačním diagramem, diagramem přehledu interakcí a diagramem časování náleží sekvenční diagram do skupiny interakčních diagramů (interaction diagrams). Diagramy interakce, popisující interakce mezi elementy, náleží do modelů znázorňujících dynamické chování systému. Během realizace případů užití se pak používají k modelování interakcí mezi objekty provádějícími konkrétní případ užití nebo jeho části [5].

Sekvenční diagram objasňuje interakci mezi objekty vyjádřenou posloupností zasílání zpráv. Sekvenční diagram má dvě dimenze: horizontální představující jednotlivé čáry života (objektů), vertikální pak čas plynoucí ve směru shora dolů. Diagram zachycuje pouze posloupnost zasílaných zpráv a časová dimenze není vázána na reálný čas.



Obr. 1.7 Sekvenční diagram.

Elementy zachycené v diagramu na obrázku 1.7:

- **Čára života** (lifeline) – reprezentuje účastníka v interakci, kterým většinou bývá objekt (v obrázku Aktér1, Objekt1-2). Lifeline také ukazuje délku života účastníků. Ve výše uvedeném obrázku účastníci v době iterace již existovali a po skončení dále existovat budou. K vyznačení vzniku a zániku objektu využíváme operace „create“ a „destroy“. Zvýrazněný obdélník na čáře života demonstruje aktivitu objektu v interakci (perioda zpracování včetně podřízených objektů).
- **Zpráva** – zajišťuje komunikaci mezi objekty, které reagují na podněty zaslaných zpráv. Reakcí může být vytvoření objektu, spuštění operace nebo odeslání signálu. Zprávy rozdělujeme na asynchronní a synchronní. Asynchronní zprávou objekt předá informaci příjemci a dál pokračuje v provádění algoritmu, aniž by čekal na dokončení úlohy příjemcem. Synchronní zpráva zaručuje čekání odesílatele na dokončení úlohy

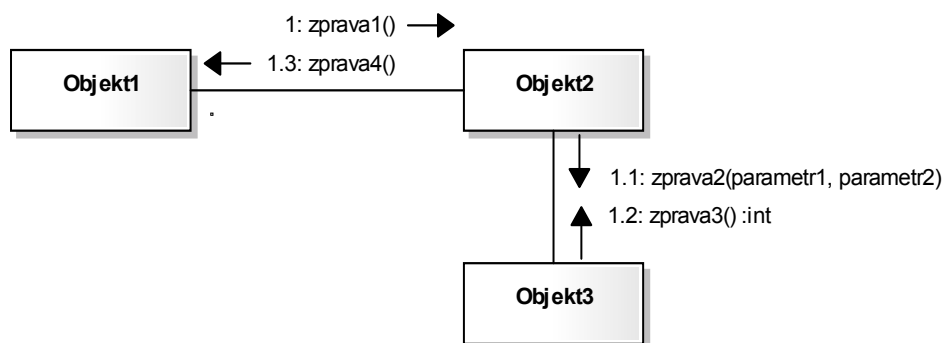
příjemcem. Návrátové zprávy (result) nejsou povinné, ale pomáhají ke zpřehlednění diagramu. Mezi další neméně důležité vlastnosti zpráv patří přiřazení parametrů a návratových hodnot.

- **Fragment** – novinka v sekvenčních diagramech zavádějící možnost rozdělení diagramu na oblasti s odlišným chováním. Například fragmentem „alt“ a „opt“ zavádíme podmíněné větvení, „loop“ zajišťuje opakování, „break“ ukončuje posloupnost atd.

### 1.2.2.6 Komunikační diagram (Communication diagram)

Tento diagram je možné v UML 1.x nalézt také pod názvem kolaborační diagram. Pojem Collaboration byl totiž ale zavádějící, jelikož se význam přibližoval statické struktuře.

Komunikační diagram, vyskytující se jak v analýze, tak v návrhu, zachycuje množinu spolupracujících objektů, které si mezi sebou zasílají zprávy. Z tohoto hlediska se diagram spolupráce obsahově podobá sekvenčnímu diagramu a zobrazuje stejný problém z jiného pohledu. Diagram spolupráce navíc nad rámec obsahu sekvenčního diagramu explicitně popisuje statickou strukturu spolupracujících objektů (obrázek 1.8). Za pomoci nástrojů je možné tyto dva diagramy automatizovaně mezi sebou převádět, avšak bez použití rozšiřujících prvků v sekvenčních diagramech.



Obr. 1.8 Komunikační diagram.

Syntaxe komunikačních diagramů je obdobná sekvenčním diagramům. Uzly zastupující objekty znázorňují účastníky v interakci. Čáry spojující uzly znázorňují komunikační cesty, jež mohou mít své vlastnosti jako je název, násobnost, role objektů. Zprávy zajišťují komunikaci mezi objekty a podobně jako u sekvenčních diagramů se dělí na tři typy (synchronní, asynchronní, návratové) a mohou obsahovat parametry a návratový typ. Navíc jsou zprávám přidělena pořadová čísla vyjadřující jejich časovou posloupnost.

### 1.2.2.7 Ostatní diagramy

Neméně důležité součásti UML, které však nejsou předmětem této práce, a proto jsou níže pouze stručně popsány:

- **Stavový diagram (State Chart)** – zachycuje stavy objektu, kterými prochází v průběhu životního cyklu programu. Ke změně stavu dochází konkrétní událostí nebo také jednoduchým plynutím času. Každý objekt má svůj počáteční (inicializační) stav a může také dojít do konečného stavu.
- **Diagramy interakce (Interaction):**
  - **Diagram časování (Timing)** – používá se k zobrazení interakcí, kde hlavním záměrem je zobrazit podrobnosti o času. Jedná se o odlišnou formu sekvenčního diagramu zobrazující změny stavu objektů v čase (v daných časových jednotkách).
  - **Stručný diagram interakce (interaction overview)** – kombinuje diagramy interakcí s diagramem aktivit, což přináší možnost sloučit výhody obou typů (využití elementu a syntaxe modelů) a modelovat tak v hlavním diagramu aktivit všechny vzájemné interakční souvislosti.
- **Diagram komponent (Component)** – komponentový diagram popisuje strukturu realizace softwarového systému zobrazenou pomocí softwarových komponent, rozhraní a jejich vzájemných závislostí.
- **Diagram nasazení (Deployment)** – znázorňuje topologii softwarových prvků nasazenou do fyzické struktury hardwarových prvků společnosti včetně různých vazeb mezi nimi.
- **Diagram složené struktury (composite structure)** – ukazuje propojené prvky systému za běhu, které spolupracují k dosažení cíle. Zachycuje statickou strukturu beze zpráv.
- **Diagram balíčků (Packages)** – balíček je ve skutečnosti univerzálním mechanismem uspořádání prvků a diagramů do skupin. Balíčky umožňují tvorbu průchodných a jasně strukturovaných modelů, neboť umožňují seskupovat předměty s těsnými sémantickými (významovými) vazbami [1].

### 1.3 Důvody výběru popsaných nástrojů

Pro tvorbu návrhu, který je obsahem této práce, byla vybrána specifikace požadavků a jazyk UML.

Specifikace požadavků představuje silný nástroj pro odhalení všech uživatelů požadovaných funkcionalit a vlastností. Již v průběhu lze vytipovat hlavní požadavky, které specifikují podstatné parametry systému, jimž bude přisouzena vyšší váha při návrhu i vytváření systému.

UML byl zvolen ze tří hlavních důvodů, díky kterým se přehoupala váha rozhodování na jeho stranu. První důvod spočívá v univerzálnosti nástroje poskytujícího bohatou škálu využitelných prvků a nepopíratelné přednosti zdůrazněné již v kapitole 1.2.

Navrhovaná aplikace je určena pro jedno menší oddělení mezinárodní společnosti využívající moderní, sofistikované metodiky a nástroje při vývoji softwarových aplikací a optimalizaci procesů. Proto také jazyk UML spadá mezi nástroje využívané v této společnosti, v čemž spatřujeme druhý a hlavní důvod výběru tohoto řešení.

Třetím důvodem není nic jiného než vizuálně přehledná podoba UML diagramů umožňujících v mnoha ohledech jednoduché a přesné vyjádření modelované oblasti. Proto i člověk neznalý syntaxe velmi rychle pronikne do problému a dokáže z diagramů číst.

## 2 Analýza současného stavu

### 2.1 O společnosti

Společnost **GE** (*General Electric*) byla založena před více než sto lety v roce 1892. U její kolébky stál geniální vynálezce Thomas Alva Edison. Dnes je GE široce diverzifikovaným koncernem, jehož činnost sahá od výroby leteckých motorů, přes lékařskou techniku, infrastrukturní řešení, přepravu, telekomunikace či stavebnictví až po poskytování finančních služeb. Součástí koncernu je i přední americká televizní společnost NBC Universal [10].

Po celou dobu své existence je GE synonymem kvality a špičkových technologií. Po několik let za sebou se stala nejobdivovanější společností světa. Vysvědčením kvality a zdraví firmy je její ratingové ohodnocení nejvyšším stupněm AAA. Z hlediska tržní kapitalizace patří GE mezi tři největší společnosti na světě. Je přítomna ve více než sto zemích světa a zaměstnává přes 315 tisíc pracovníků [10].

Významnou součástí finanční divize GE je skupina **GE Money**. Ta spravuje aktiva ve více než 50 zemích a zaměstnává cca 60 tisíc lidí. Je předním poskytovatelem finančních služeb pro 130 milionů klientů - jednotlivců i malých a středních podniků. K produktům a službám, které poskytuje, patří osobní půjčky a podnikatelské úvěry, platební a úvěrové karty, konsolidace úvěrů, hypotéky, refinancování hypoték, depozita, investiční produkty, úvěrové pojištění, leasing automobilů, ale i poradenství v oblasti získávání veřejné podpory na vybrané typy projektů [10].

V České republice společnost GE Money úspěšně působí již od roku 1998. V rámci podpory zákaznických služeb zde GE vybudovala dvě zákaznická centra se sídly v Praze a Ostravě.

## 2.2 Objekt řešení

Objektem řešení této práce je incentivní (prémiový) systém operátorů oddělení *Akvizic* výše zmíněné společnosti. Oddělení se nachází v centru zákaznických služeb v Ostravě. Akvizice se dělí na několik týmů, kde každý zabezpečuje specifický interní proces (z důvodu zachování know-how společnosti není možno na tomto místě dále rozvádět firemní procesy a postupy).

Reakcí na dění v ekonomice a peněžnictví dochází na oddělení, stejně jako u jiných společností, k organizačním, funkčním nebo procesním změnám. Změny, aby byly trvale a obvykle rychle zakomponovány do interních procesů, musí být jednak dobře komunikovány zaměstnancům, avšak také zaměstnanci k nim musí být trvale motivováni. Proto vznikla v popisovaném případě potřeba vytvoření incentivního systému, jenž by automatizovaně reagoval na změny (případně s drobným zásahem vedoucího týmu), zpracovával výkony operátorů dle nastavených pravidel a vložených, respektive importovaných dat, a posléze vyhodnotil jejich prémiové nároky. Pod pojmem incentivní systém se tedy skrývá jak sofistikovaný univerzální vzorec spravedlivě poměřující výkony operátora pracujícího na více procesech, tak aplikace spravující pravidla, data o výkonech, operátorech, prémiech a zabezpečující import a export dat.

Předmětem této práce je právě zmiňovaná aplikace, jejímž vytvořením zefektivníme proces výpočtu výkonů a zautomatizováním některých činností ušetříme čas obětovaný pro měsíční agendu výkonů a prémie. Samozřejmě není opomenuta ani revize procesu a nastavení nových pravidel (univerzální vzorec), což bude zmíněno v kapitole 3.1. Nutno ale dodat, že tato problematika není prioritně obsahem této práce.

## 2.3 Současný stav

Jak již bylo zmíněno oddělení Akvizic se skládá z několika týmů, za jejichž fungování odpovídají pověření vedoucí (Team Leaderi – TL). Každý vedoucí v současnosti samostatně zpracovává agendu incentivního systému pro svůj tým, v čemž je zahrnut časově náročný sběr informací, výpočet výkonů dle nastavených pravidel a přerozdělení prémie. K tomuto procesu využívají interních databází a jednoduchých nástrojů z kancelářského balíku MS Office.

Na počátku byl výše uvedený postup plně dostačující, jelikož každý tým vykonával jasně danou stejnorodou činnost, což s připravenými šablonami až tak časově nezatěžovalo vedoucího týmu. S reorganizací a postupným vývojem společnosti došlo k mnoha změnám



ve vykonávaných činnostech jednotlivých týmů ve smyslu rozšíření množství a druhu vykonávaných činností. To podstatně zkomplikovalo pravidla a celý výpočet výkonů. K složitosti výpočtů přispělo také nastavení odlišných pravidel u každého týmu, čemuž se z důvodu zpracovávání různorodých činností nedalo vyhnout. Z druhého hlediska - z pohledu vedoucího, přestože zaměstnanec vykonává pravidelně opakovanou činnost, při dané výši složitosti se zvyšuje pravděpodobnost vzniku chyb, kterým se při manuálním zpracovávání dá jen stěží vyhnout. Tato situace dospěla až do fáze, kdy přehodnocení procesu rozdělování incentiv a vytvoření aplikace pro správu výkonů se stalo nevyhnutelným krokem pro efektivní práci vedoucího týmu, manažera i operátorů.

### 3 Návrh a realizace řešení

Modelovaný systém by měl plnit několik hlavních funkcí. Jeho první role spočívá v evidenci důležitých dat potřebných pro správu a výpočet výkonů. Jelikož se jedná o citlivá osobní data, není možné při tvorbě aplikace opomenout zákon o ochraně osobních údajů. Evidence všech výkonových dat oddělení v jednom datovém uložišti přináší přehlednost, možnost vytěžování nových dat a tvorbu nových pravidelných reportů. Další funkcionalitou systému bude automatizace určitých činností procesu šetřící časové nároky uživatelů. Mezi nejvýznamnější vlastnosti aplikace řadíme univerzálnost umožňující rychlé změny dle aktuálních potřeb oddělení společnosti a společný flexibilní algoritmus výpočtu pro všechny operátory zajišťující spravedlivější výsledek. Podrobné funkcionality aplikace jsou popsány ve specifikaci požadavků (viz kapitola 3.2).

Systém by neměl splňovat pouze výše popsané charakteristiky, ale v současnosti je již samozřejmostí určitý uživatelský komfort a intuitivní navigace v rámci interakce uživatele se softwarovým prostředím. Dále nesmí být opomenuty přehledné výstupy v textové i grafické podobě umožňující dynamický výběr reportovaných výsledků.

Jak vyplývá z předchozího textu, nejvíce zapojenými uživateli budou vedoucí týmů, pro které je v podstatě systém vytvářen. Do systému však budou vstupovat také operátoři (zadávatelé odpisové hodiny a sledující své výkony) a manažeři oddělení (pro něž budou sloužit souhrnné přehledy výkonů a přehled čerpaní budgetů na výplatu prémie).

Z důvodu optimalizace procesu a nasazení nového algoritmu výpočtu výkonů a prémie, je nezbytné si nejdříve celý proces nasimulovat pro prověření všech správných funkcí a závislostí následných kroků, otestovat algoritmus pro zjištění výsledků nového měření a porovnat jej se staršími výsledky. Pro simulaci procesu a testování algoritmu bylo zvoleno řešení vytvořit prototyp aplikace, splňující ty nejdůležitější procesní požadavky.

### 3.1 Optimalizace procesu a algoritmus výpočtu

Před samotným začátkem návrhu aplikace nesmí být opomenuta fáze revize a optimalizace procesu, protože není efektivní navrhovat aplikaci pro proces, ve kterém se mohou vyskytovat nedostatky, nebo dokonce pro nedokonale fungující proces. S analyzovaným procesem úzce souvisí algoritmus výpočtu výkonů a prémie, který bylo zapotřebí přizpůsobit aktuálním podmínkám a vývoji oddělení. Revize a optimalizace probíhala ve spolupráci vedoucích týmů (TL), manažer oddělení a samozřejmě analytika.

V první fázi revize byl kompletně zmapován opakující se proces výpočtu incentiv probíhající v pravidelných měsíčních intervalech a ve všech týmech. Během mapování byly také shromažďovány všechny potřebné a dostupné informace o této problematice, které napomohou při následné optimalizaci procesu.

Po zmapování procesu byl uspořádán první workshop, kde se přednesly návrhy a připomínky všech účastníků. Postupnou analýzou všech následných kroků a rozбором možných změn byly odhaleny eventuální problémové akce a návaznosti procesu. Zavedení změny do procesu bylo podmíněno souhlasem všech výše zmíněných účastníků. Následně byly změny zapracovány do procesu a v teoretické rovině simulovány všechny možné varianty, které by mohly nastat při průchodu procesem. Během simulace došlo k odhalení drobných nedostatků vyskytujících se sice zřídka, ale ohrožujících průběh procesu. Tyto nedostatky byly samozřejmě analyzovány a proces upraven, aby nemohlo dojít k nežádoucím stavům.

Jak již bylo zmíněno v úvodu této kapitoly, vznikla potřeba přizpůsobit algoritmus výpočtu incentiv dle aktuálních požadavků oddělení. Před započítáním akcí realizujících vytvoření nového vzorce výpočtu se odsouhlasil hlavní záměr realizace algoritmu, a to vytvoření komplexního, flexibilního a přehledného algoritmu neomezujícího při vývoji oddělení a umožňujícího výpočet výkonu pracovníka, který bude zpracovávat jakoukoliv množinu činností daného oddělení. Zmiňovaná komplexnost spočívá ve spojení výkonu za činnosti, jež jsou poměřovány různými stupni kvality a kvantity. Uspořádáním dalších workshopů se podařilo nalézt optimální algoritmus splňující požadované podmínky, přičemž byla zachována přehledná a srozumitelná podoba algoritmu. Opět nebyla opomenuta simulace výpočtu dle nového algoritmu odhalující možné problémy. Po simulaci byly porovnány výsledky a všemi účastníky schváleno nasazení nového algoritmu.

## 3.2 Specifikace požadavků

Při hledání požadavků na systém byly využity techniky přímé komunikace s uživatelem (otevřený rozhovor, kolektivní přijímání rozhodnutí – brainstorming a workshop), pozorování a analýza cílů. V tabulce 3.1 jsou vypsány všechny systémové požadavky. Tabulka 3.2 definuje nastavení stupnice priority stanovující důležitost požadavků. Podle priority bude přístupováno k požadavkům při jejich implementaci do systému.

### 3.2.1 Seznam požadavků:

ID	Popis	Priorita	Druh
R <sub>1</sub>	Evidence zaměstnanců.	1	F
R <sub>2</sub>	Evidence pravidel.	1	F
R <sub>3</sub>	Evidence typů odpisů.	1	F
R <sub>4</sub>	Evidence budgetů.	1	F
R <sub>5</sub>	Omezení přístupu k aplikaci dle uživatelských práv.	2	F
R <sub>6</sub>	Zobrazování informací v aplikaci dle uživatelských práv.	2	F
R <sub>7</sub>	Evidence záznamů odpisů.	1	F
R <sub>8</sub>	Zajištění procesu schvalování odpisů.	1	F
R <sub>9</sub>	Napojení na externí data – import dat.	1	F
R <sub>10</sub>	Evidence výkonů.	1	F
R <sub>11</sub>	Automatický výpočet výkonů.	1	F
R <sub>12</sub>	Efektivní rozdělování prémie dle výkonu a nastavení.	2	F
R <sub>13</sub>	Možnost manuálního přerozdělení prémie.	1	F
R <sub>14</sub>	Online sledování čerpaných budgetů.	2	F
R <sub>15</sub>	Proplacení prémie účtárnou.	2	F
R <sub>16</sub>	Hlavní přehled výkonů, prémie a budgetů.	2	F
R <sub>17</sub>	Detailní přehled budgetů a zůstatků.	3	F
R <sub>18</sub>	Detailní přehled výkonů.	3	F
R <sub>19</sub>	Detailní přehled schválených odpisů.	3	F
R <sub>20</sub>	Přehled osobních výkonů operátora.	2	F
R <sub>21</sub>	Export přehledů.	3	F
R <sub>22</sub>	Uživatelsky příjemné prostředí.	2	N
R <sub>23</sub>	Vzhled aplikace přizpůsobený firemním barvám.	2	N
R <sub>24</sub>	Víceuživatelské prostředí.	1	N
R <sub>25</sub>	Rychlost odezvy aplikace.	2	N

Tab. 3.1 Seznam požadavků.

Písmena N a F v tabulce 3.1 znamenají rozdělení na funkční a nefunkční požadavky.

Priorita	Název	Popis priority
1	Nutné	Jádro problému, nezbytné pro chod aplikace.
2	Možné	Velmi důležité požadavky, které však lze vynechat.
3	Chceme	Nepovinné požadavky, avšak zdokonalující využitelnost aplikace.

Tab. 3.2 Definice priorit.

### 3.2.2 Popis požadavků

**R1 Evidence operátorů** - Oddělení dlouhodobě sleduje vývoj svých zaměstnanců (operátorů a vedoucích). U každého zaměstnance jsou vedeny jeho základní atributy, z nichž některé slouží jako jednoznačné identifikátory a jiné k evidenci a sledování historických změn (např. jméno, příjmení, plat atd.). Sledování historických změn je rovněž podstatné pro monitorování kariérního vývoje zaměstnance. U každého zaměstnance evidujeme také práva omezující přístupy k aplikaci.

**R2 Evidence pravidel** - Při výpočtu výkonů využíváme variabilní vzorec, který lze upravovat pomocí tzv. pravidel. Tato pravidla se mohou lišit po jednotlivých obdobích a povolení k jejich změně mají pouze vedoucí týmů. Mezi pravidla patří nastavení normočasů u jednotlivých činností a dále váhy přisuzující význam mezi výkonem a kvalitou.

**R3 Evidence typů odpisů** - Odpis – vžitý název činnosti, která spadá do náplně práce zaměstnance a prozatím ji nelze měřit systémově. Odpis se ve své podstatě rovná počtu hodin strávených na určité aktivitě (neměřitelné činnosti) a tento počet je následně odečten z disponibilního časového fondu zaměstnance (což zvyšuje jeho pracovní výkon). Proto je vytvořen číselník odpisů (např. školení, rotace, práce pro vedoucího apod.) pomáhající specifikaci odpracovaných činností a následným statistikám.

**R4 Evidence budgetů** - Budgety (rozpočty) představují sumu finančních prostředků určenou pro výplatu prémie. Tato evidence slouží k porovnávání rozpočtu s aktuálními finančními toky plynoucími na prémie.

**R5 Omezení přístupu k aplikaci dle uživatelských práv** - Aplikace musí automaticky vyhodnotit uživatele a podle jeho práv umožnit nebo zamítnout přístup. Toto vyhodnocení provede aplikace pomocí jedinečného uživatelského jména získaného z operačního systému a hesla zadaného uživatelem.

**R6 Zobrazování informací v aplikaci dle uživatelských práv** - Každý uživatel aplikace má přidělena omezující práva. Aplikace vyhodnotí práva uživatele a zobrazí informace podle jeho aktuálně nastavených práv. Porušení práv a pokusy o nabourání systému budou zaznamenávány do relace s logovacími informacemi.

**R7 Evidence záznamů odpisů** - Operátor přidává a upravuje záznamy odpisu. Vybírá je z číselníku typů odpisů, zadává počet hodin a datum vykonání. Jelikož tyto záznamy mají vliv na výkon operátora, probíhá po zadání záznamu proces schvalování vedoucím (viz R<sub>8</sub>). Schválené záznamy jsou viditelné v přehledech a zamítnuté operátor maže nebo dále upravuje a posílá znovu ke schválení.

**R8 Zajištění procesu schvalování odpisů** - Vedoucí schvaluje všechny záznamy vytvořené operátory spadající do jeho působnosti. Každý záznam porovnává se skutečností a schvaluje, případně zamítá. U zamítnutí musí zadat důvod zamítnutí, aby se operátor dozvěděl příčinu a mohl patřičně reagovat.

**R9 Napojení na externí data – import dat** - Informace týkající se výkonů operátorů vede společnost ve sdílených databázích. Online napojením aplikace na rozsáhlé firemní databáze by aplikace nesplňovala požadavek rychlé odezvy. Proto aplikace musí zabezpečovat import agregovaných dat na operátora, a to pravidelně za předem stanovené období.

**R10 Evidence výkonů** - Určité informace k výpočtu výkonů se z mnoha důvodů budou zadávat manuálně (části týkající se vyšší bezpečnosti a podobně).

**R11 Automatický výpočet výkonů** - Po importu dat a po manuálním zadání dat aplikace vyhodnotí vstupy a podle zadaného vzorce (vzorec a nastavení) automaticky vypočte výkony operátorů. Tento výpočet se flexibilně mění dle aktuálních vstupních dat.

**R12 Efektivní rozdělování prémie dle výkonu a nastavení** - Vedoucí týmu má určité pravomoci jak nakládat se svým rozpočtem na období. Proto musí být v aplikaci funkcionality flexibilního nastavení umožňující rozdělení prémie odpovídající výkonu operátorů a parametrům vedoucího (např. limit rozpočtu na období).

**R13 Možnost manuálního přerozdělení prémie** - Všechny aktivity pracovníka nelze vyjádřit kvantitativně, a proto aplikace musí umožňovat manuální doplnění prémie (může se např. jednat o odměnu za aktivity nad rámec práce, dlouhodobě vysoké pracovní výkony, ale v tomto přerozdělení se také vyskytují srážky odrážející chování zaměstnance neslučující se s pravidly společnosti).

**R14 Online sledování čerpaných budgetů** - Aplikace bude zobrazovat aktuální stav budgetu a v době výpočtu a přerozdělování prémie automaticky přepočítávat zůstatky.

**R15 Proplacení prémie účtárnou** - Po přerozdělení prémie zadá vedoucí týmu pokyn k vyplacení. Po tomto pokynu jsou data předána ke zpracování účtárně.

**R16 Hlavní přehled výkonů, prémie a budgetů** - Tento hlavní přehled zobrazí hlavní agregované výsledky týmu/oddělení (např. průměrný výkon, aktuální stavy budgetů atd).

**R17 Detailní přehled budgetů a zůstatků** - Přehled zobrazující aktuální stavy budgetů v detailním rozpadu po jednotlivých týmech, operátorech a obdobích.

**R18 Detailní přehled výkonů** - Přehled zobrazující výsledky výkonů v detailním rozpadu po jednotlivých týmech, operátorech a obdobích.

**R19 Detailní přehled schválených odpisů** - Přehled zobrazující množství schválených odpisů (včetně počtu hodin a MD) v detailním rozpadu po jednotlivých týmech, typech odpisů, operátorech a obdobích.

**R20 Přehled osobních výkonů operátora** - Přehled zobrazující výsledky výkonů operátora v detailním rozpadu po jednotlivých obdobích.

**R21 Export přehledů** - Funkcionalita aplikace umožňující export dat z aktuálně zobrazeného přehledu. Export převede data do formy tabulky ve formátu MS Excel.

**R22 Uživatelsky příjemné prostředí** - Pod tímto požadavkem se skrývá vzhled aplikace (použití barev zvýrazňujících přehlednost, ale zároveň příjemných pro oko), uživatelsky intuitivní vzhled (použití stejných názvů pro tutéž funkcionalitu, použití takových názvů funkcionalit intuitivně směřujících uživatele k dané akci, podobné rozmístění prvků ve formulářích, celková přehlednost nabídek apod).

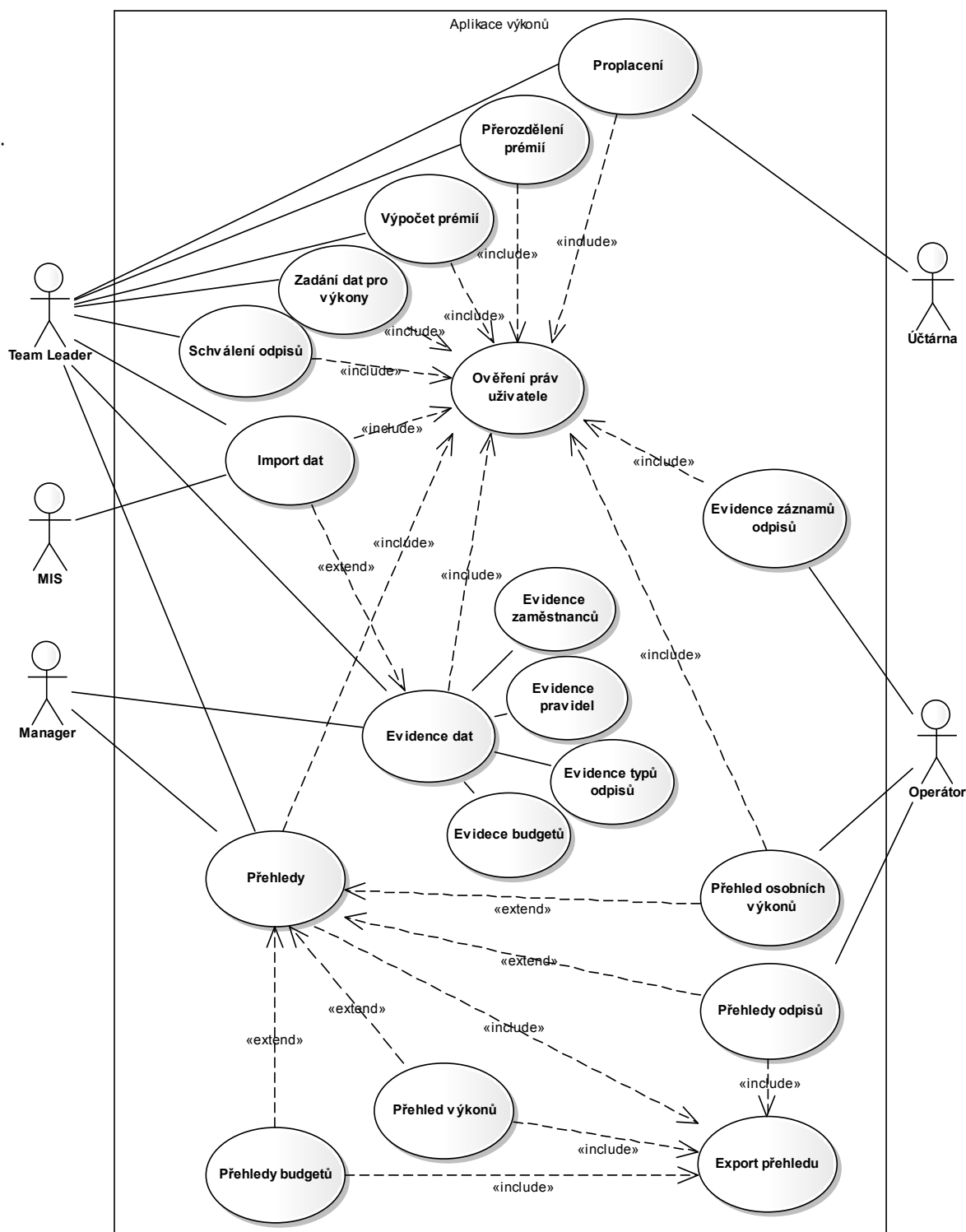
**R23 Vzhled aplikace přizpůsobený firemním barvám** - Využití takových barev v aplikaci (nabídky, formuláře, tlačítka) by mělo korespondovat s korporátní preferovanou škálou barev.

**R24 Víceuživatelské prostředí**- Aplikace musí zabezpečit víceuživatelský přístup (v jeden okamžik bude využívat aplikaci více uživatelů).

**R25 Rychlost odezvy aplikace** - Využití takové technologie zajišťuje téměř okamžitou odezvu aplikace (např. do 2 sekund).

### 3.3 Modelování případu užití

Níže uvedený diagram případu užití (obr. 3.1) znázorňuje budoucí podobu systému zpracováním funkčních potřeb aktérů. Z diagramu jsou zřejmé vztahy mezi aktéry a případy užití a hranice modelovaného systému.



Obr. 3.1 Diagram případu užití.



Každý případ užití by měl mít pro přehlednost specifický název, jenž by odrážel účel vybraného případu užití, a přiřazeno unikátní ID, kterým budou jednoznačně identifikovány UC při následných fázích modelování.

ID	popis	ID	popis
UC1	Evidence zaměstnanců	UC10	Přerozdělení prémie
UC2	Evidence pravidel	UC11	Proplacení
UC3	Evidence typů odpisů	UC12	Přehledy
UC4	Evidence Budgetů	UC13	Přehledy budgetů
UC5	Import Dat	UC14	Přehledy odpisů
UC6	Evidence záznamů odpisů	UC15	Přehledy výkonů
UC7	Schválení odpisů	UC16	Přehledy osobních výkonů
UC8	Zadání dat pro výkony	UC17	Export přehledu
UC9	Výpočet prémie	UC18	Ověření práv uživatele

Tab. 3.3 Seznam případů užití.

### 3.3.1 Specifikace účastníků

**Manager** – vedoucí zodpovědný za chod celého oddělení (spravuje seznam budgetů, sleduje veškeré přehledy – převážně přehledy výkonů a čerpání budgetů).

**Team Leader** – vedoucí týmu operátorů, který zabezpečuje provoz určitého procesu (nebo množiny procesů). Tento účastník bude nejfrekventovanějším uživatelem aplikace, a to k evidenci všech potřebných dat, výpočtu výkonů, přerozdělení prémie a reportování výsledků týmu.

**Operátor** – název pro účastníky pracující v určitých týmech (zadávací odpisové hodiny, sledující osobní výkony a schválené odpisy).

**MIS** – systém databází společnosti. Zdroj dat k evidenci zaměstnanců, budgetů a hlavně výkonů operátorů (z tohoto zdroje získáváme agregované počty zpracovaných činností na operátora a období).

**Účtárna** – oddělení společnosti zajišťující vedení účetnictví a vyplacení finančních transakcí (pro systém podstatné, že přijímají data k vyplacení prémie).

### 3.3.2 Matice RTM

Pro ověření souladu mezi požadavky a nadefinovanými případy užití byla vytvořena matice sledovanosti požadavků (RTM). Matice dokazuje pokrytí všech požadavků v rámci případů užití, čímž bylo prověřeno neopomenutí žádného požadavků v rámci UC modelu.

		Případy užití																	
Požadavky		UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18
	R <sub>1</sub>																		
	R <sub>2</sub>																		
	R <sub>3</sub>																		
	R <sub>4</sub>																		
	R <sub>5</sub>																		
	R <sub>6</sub>																		
	R <sub>7</sub>																		
	R <sub>8</sub>																		
	R <sub>9</sub>																		
	R <sub>10</sub>																		
	R <sub>11</sub>																		
	R <sub>12</sub>																		
	R <sub>13</sub>																		
	R <sub>14</sub>																		
	R <sub>15</sub>																		
	R <sub>16</sub>																		
	R <sub>17</sub>																		
	R <sub>18</sub>																		
	R <sub>19</sub>																		
	R <sub>20</sub>																		
	R <sub>21</sub>																		
	R <sub>22</sub>																		
	R <sub>23</sub>																		
	R <sub>24</sub>																		
	R <sub>25</sub>																		

Tab. 3.4 RTM matice sledovanosti požadavků.

### 3.3.3 Specifikace případu užití

Jak již bylo popsáno v kapitole 1.2.3, UML přesně nestandardizuje podobu a náležitosti specifikace případu užití. Pro tuto práci byla využita a mírně upravena šablona z knihy *UML2 a unifikovaný proces vývoje aplikací* [1]. Tvorba specifikací probíhala ve spolupráci se zaměstnanci společnosti (uživatelé budoucí aplikace), aby byly zachyceny všechny požadavky, podstatné souvislosti a parametry. Podobu specifikace případu užití představuje tabulka 3.5. Všechny kompletní specifikace jsou zaznamenány v příloze č.1.

<b>Případ užití: Evidence zaměstnanců</b>
<b>ID: UC1</b>
<b>Primární aktéři:</b> Team leader Manager <b>Vedlejší aktéři:</b> Operátor
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Operátor poskytne osobní data.
<b>Hlavní scénář:</b> 1. Seznam zaměstnanců je spuštěn primárním aktérem. 2. Když PA přidává záznam: 2.1 PA iniciuje vytvoření záznamu. 2.2 PA vkládá data. 3. Když PA edituje záznam: 3.1 PA vyhledá záznam. 3.2 PA edituje záznam. 4. Primární aktér zadá pokyn k uložení záznamu. 5. Záznam zaměstnance je uložen do databáze.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje evidenci. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží záznam a zobrazí informaci o uložení. 2. Změny v evidenci se projeví v celém systému.

Tab. 3.5 Specifikace případu užití – evidence zaměstnanců (PA- primární aktér).

### 3.4 Analýza případu užití

Během této fáze byly analyzovány materiály vytvořené během modelování případů užití a při specifikaci požadavku. Byl vytvořen analytický model tříd zobrazující statickou strukturu systému sloužící jako podklad k implementaci systému.

#### 3.4.1 Model tříd

Tento model tvoří podklad pro další fázi realizace případů užití. Při modelování chování systému totiž pracujeme s jednotlivými objekty, u kterých přidáním určitých prvků znázorňujeme chování a modelujeme vzájemné interakce. Výstupem této fáze je kandidátka tříd, atributů, odpovědností (tab. 3.6) a model analytických tříd (obr. 3.2).

K vytvoření modelu analytických tříd byla vybrána technika „analýzy podstatných jmen a sloves“. Pro použití této techniky byly vybrány informace definované ve specifikaci požadavků a dále z materiálů poskytnutých společností (popis procesů, zápisy z workshopů). Následující analyzovaný text byl vytvořen výběrem podstatných částí z dostupných materiálů. Vybraná podstatná jména a spojení podstatných jmen jsou v textu označeny podtržením a vybraná slovesa se slovesnými frázemi kurzívou.

##### 3.4.1.1 Analyzovaný text

Na oddělení *nastoupí* nový pracovník. Team leader (vedoucí týmu - dále jen TL) nebo manager (dále jen MNG) *zaeviduje* daného pracovníka do systému. Určité informace *poskytne* pracovník (jméno, příjmení, telefon, email) a zbylé *doplní* TL ( ID pracovníka, ID oddělení, ID týmu, plat, datum nástupu, případně datum odchodu).

Pracovník *spadá* do těchto kategorií: operátor, TL (*zodpovídá za* seznam procesů) a MNG (*zodpovídá za chod celého oddělení =>* seznam odpovědností, *reportuje k nadřízenému oddělení*). Každý pracovník *je přiřazen* do určitého týmu zabezpečujícího vybrané procesy (název týmu, popis týmu, seznam zpracovávaných činností).

Při *spuštění aplikace uživatelem* *bude* daný uživatel automaticky *prověřen* aplikací a podle jeho práv mu *bude umožněn* (resp. *zamítnut*) přístup a *zobrazeny* (resp. *nezobrazeny*) informace. Číselník práv *eviduje* TL a správce (id práva, název, popis, dobu platnosti).

Před měřením výkonů a vyplácením prémie si TL a MNG *evidují* pravidla, která budou *platná* po určité období. Proto se nejdříve musí *nastavit platnost* období (datum od, datum

do). Poté *nastavují* jednotlivá pravidla a platnost (váhy kvality, váhy kvantity, období, tým). Speciálním typem pravidla *jsou* tzv. normočasy určující dobu strávenou na dané činnosti.

Činnost se rovná předem definované stále podobně se opakující aktivitě pracovníka mající svůj název, popis, ID týmu a dobu platnosti. Normočasy tedy souvisí s činností a obdobím.

Dále TL a MNG *evidují* budgety (rozpočty) pro všechny týmy (obsahuje období platnosti, tým, typ budgetu, výše budgetu).

V průběhu jednotlivých období si operátoři *evidují* odpisy. K vybranému odpisu *vyplňují* datum provedení odpisové činnosti, počet hodin a případně důvod. Odpis odpovídá vžitému názvu činnosti, která *spadá* do náplně práce zaměstnance a nelze ji měřit systémově. Odpis *se* ve své podstatě *rovná* počtu hodin strávených na určité aktivitě (neměřitelné činnosti) a tento počet *je* následně *odečten* z disponibilního časového fondu zaměstnance (což zvyšuje jeho pracovní výkon). Proto TL *eviduje* číselník odpisů (např. školení, rotace, práce pro vedoucího apod.), kde zaznamenává typ odpisu, popis odpisu, platnost pro tým, datum platnosti. Odpisy probíhají procesem schvalování. Při schválení odpisu TL *jsou* hodiny u odpisu *započteny* do výkonu pracovníků. Při zamítnutí TL *vyplňuje* důvod zamítnutí a záznam se vrací k editaci pracovníkovi.

Data odrážející výkony operátorů (seznam a počty činností a časy na daných úkolech) *jsou uloženy* v databázích (zkratka pro podnikové informační databáze = MIS). TL *dává* pokyn k importu dat z MISu (agregovaná data *budou* pravidelně *importována* (za dané období) z důvodu rychlosti přístupu k aplikačním funkcionalitám).

Ještě před samotným výpočtem výkonů TL *zadáva* nezbytná data (zatím nesystémově měřitelné) k výpočtu výkonů ( ID pracovníka, počet hodin v práci a mimo, výsledky testů, kvalita náslechů, kvalita ostatní práce).

Výpočet výkonů rovněž *provádí* TL spuštěním příkazu pro výpočet. Výkony a prémie *budou vypočítány* podle předem nastavených pravidel (normočasy, pravidla) a z importovaných dat. TL dále *rozděluje* ostatní prémie (ocenění za aktivity nad rámec práce, zvláštní týmové ocenění, ale také srážky za chování pracovníka neslučující se s pravidly společnosti).

V momentě přerozdělení premii TL *spouští příkaz* k proplacení (data jsou přeposlána účtárně – atributy: ID pracovníka, částka, datum proplacení).

Pro operátory *budou k dispozici* dva přehledy – přehledy osobních výkonů a přehledy schválených odpisů zobrazující se dle ID pracovníka a týmu.

Pro TL a MNG aplikace *nabízí* nejprve hlavní přehled zobrazující souhrnné informace dle jednotlivých období, týmů, typů odpisů. Rovněž *nabízí* možnost zobrazení detailních dat u každé oblasti (přehledy výkonů, budgetů, odpisů). Všechny přehledy *lze zobrazit* v tabulkové i grafické podobě.

Kandidátní třídy, atributy a odpovědnosti:

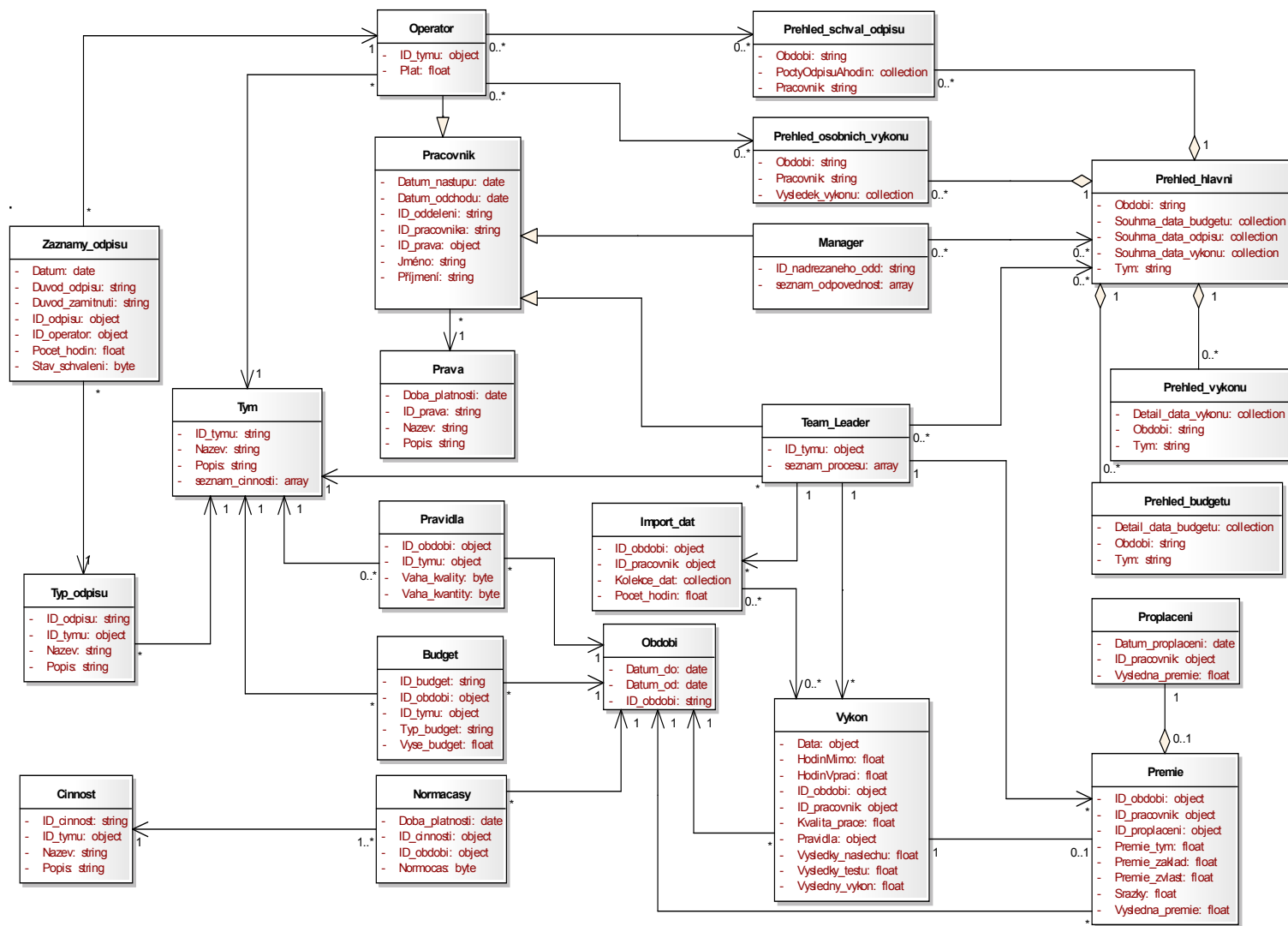
<b>Třída</b>	<b>Atributy</b>	<b>Odpovědnost</b>
Pracovník	ID pracovníka ID oddělení Jméno Příjmení Plat Datum nástupu Datum odchodu Telefon Email ID práva	Poskytne informace
Operátor	ID týmu Plat	Poskytuje osobní informace Evidence záznamů odpisů
Team leader	ID týmu Seznam procesů	Zodpovídá za procesy Evidence: pracovníků, pravidel, práv, typů odpisů, období Schvaluje záznamy odpisů Stvrzuje výpočet výkonů Rozděluje prémie Spouští import dat Evidence a výpočet budgetů Spouští příkaz k proplacení
Manager	Seznam odpovědností ID nadřazeného oddělení	Evidence: pracovníků, pravidel, práv, typů odpisů Evidence a výpočet budgetů Zodpovídá za oddělení
Tým	ID týmu Název Popis Seznam zpracovávaných činností	Určuje rozdělení pracovníků
Práva	ID práva Název Popis Doba platnosti	Umožňuje přístup Zobrazuje informace

<b>Třída</b>	<b>Atributy</b>	<b>Odpovědnost</b>
Období	ID období Datum od, Datum do	Nastavuje platnost dat
Pravidla	ID období ID týmu Váha kvality Váha kvantity	Nastavuje platnost dat Určují vlastnosti výpočtu
Činnost	ID činnosti ID týmu Název Popis	Definuje aktivitu (práci)
Normočasy	ID období ID činnosti Normočas Doba platnosti	Nastavuje platnost dat Určují vlastnosti výpočtu
Budget	ID budget ID období ID týmu Typ budgetu Výše budgetu	Určuje limity vyplacení prémie
Typ odpisu	ID odpisu ID týmu Název Popis Datum platnosti	Rozděluje odpisy do skupin
Záznamy odpisů	ID operátor ID odpisu Datum Počet hodin Důvod_odpisu Stav schválení Důvod_zamítnutí	Zpřehledňuje odpisy Zvyšuje výkon operátora
Import dat	ID pracovníka ID období Seznam s názvy a počty činností Čas na jiných činnostech	Importování dat Zrychlení práce s aplikací Agreguje výkony pracovníků
Výkon	ID pracovníka ID období Data Pravidla Počet hodin v práci Počet hodin mimo Výsledky testů Kvalita následků Kvalita ostatní práce Výsledný výkon	Seskupuje výsledky výkonů Provádí výpočet výkonů Seřazuje dle výkonů

<b>Třída</b>	<b>Atributy</b>	<b>Odpovědnost</b>
Prémie	ID pracovníka ID období ID budget ID proplacení Prémie Zvláštní prémie Týmové ocenění Srážky	Rozděluje budget do premií Aktualizuje stavy budgetů
Proplacení	ID pracovníka Celková částka Datum proplacení	Zabezpečuje proplacení premií
Přehled osobních výkonů	Pracovník Období Výsledek výkonu	Zobrazuje agregované informace
Přehled schválených odpisů	Pracovník Období Počty odpisů Počty odepsaných hodin	Zobrazuje agregované informace
Hlavní přehled	Období Tým Souhrnná data výkonů Souhrnná data budgetů Souhrnná data odpisů	Zobrazuje agregované informace
Přehled výkonů	Období Tým Detailní data výkonů	Zobrazuje agregované informace
Přehled budgetů	Období Tým Detailní data budgetů	Zobrazuje agregované informace

Tab. 3.6 Kandidátka tříd, atributů a odpovědností.





Legenda diagramu tříd	
Symbol	Popis
—	obousměrná asociace
←	jednosměrná asociace
◁	generalizace
◊	agregace
Násobnost	
0..1	nula nebo jedna
1	právě jedna
0..*	nula až mnoho
1..*	jedna až mnoho
*	mnoho

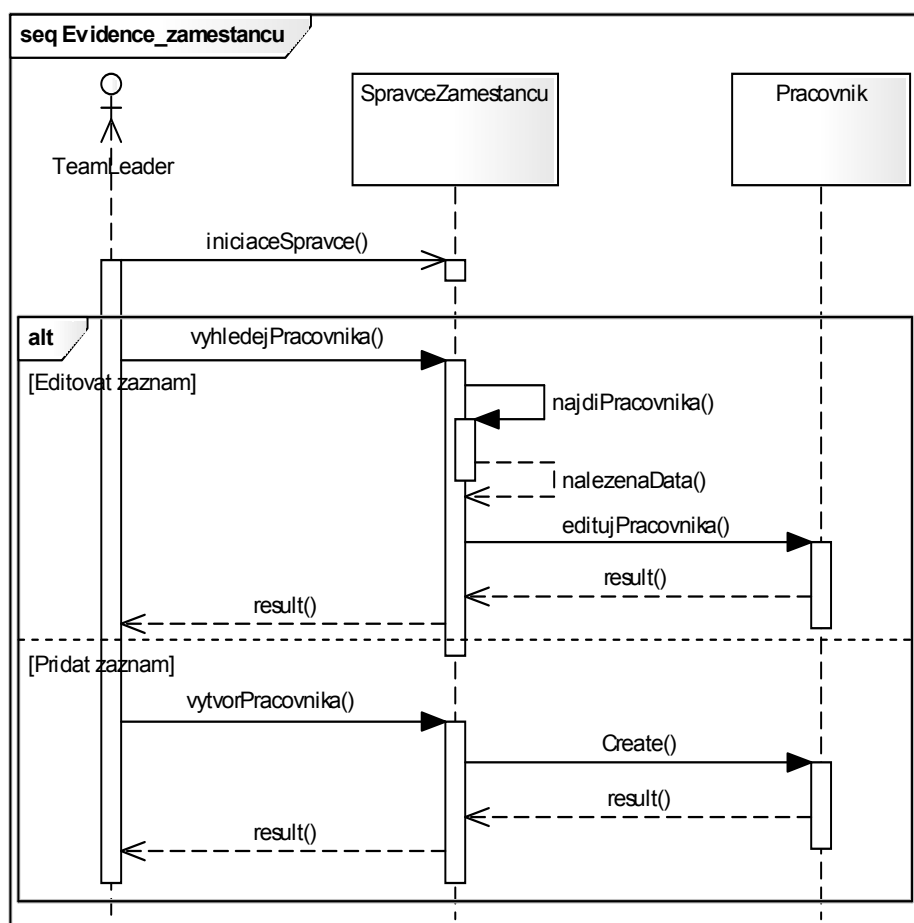
Obr. 3.2 Diagram analytických tříd.

## 3.5 Realizace případu užití – chování systému

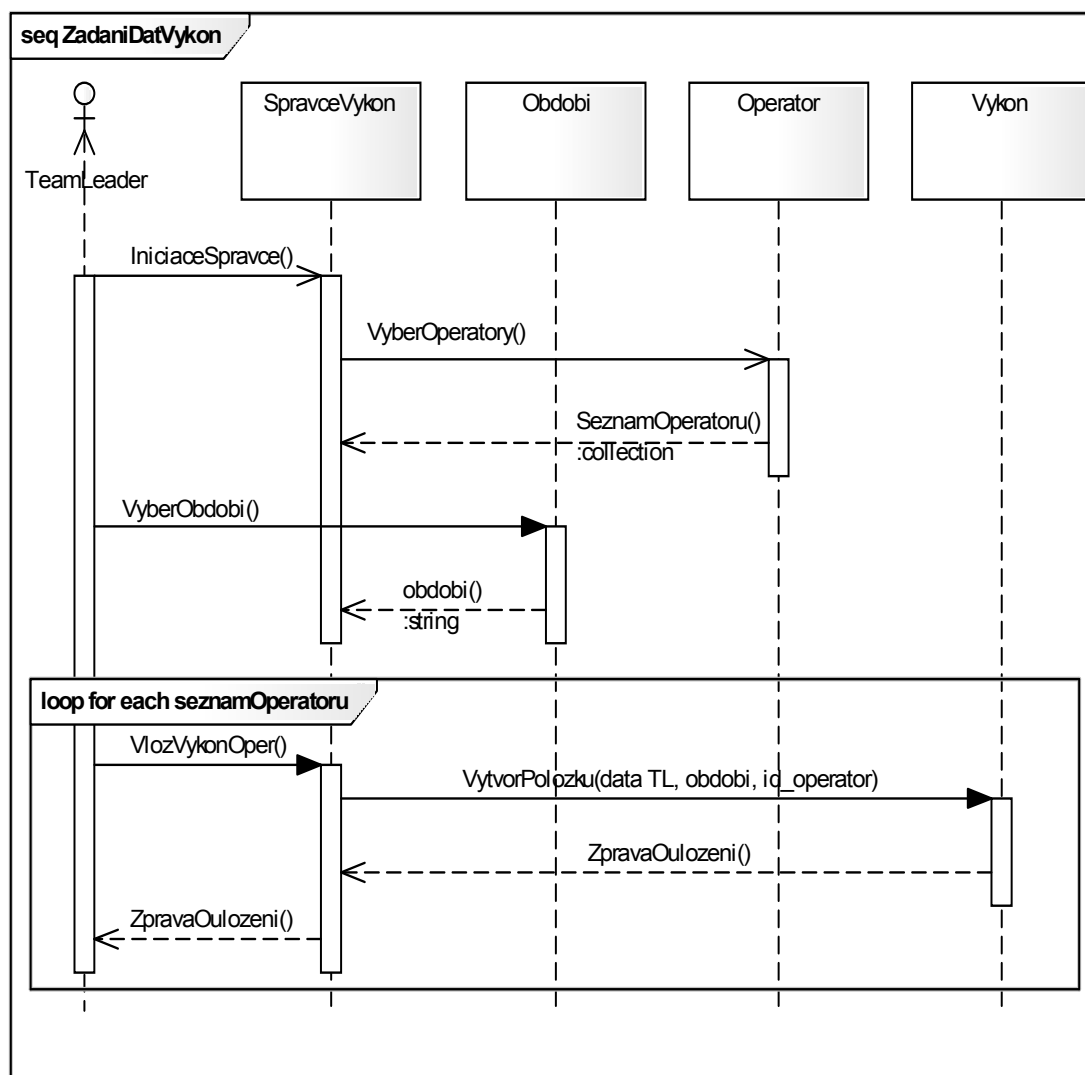
V dalším postupu budeme modelovat chování jednotlivých tříd zmíněných v předešlé kapitole a tím oživovat funkční podobu celého systému. Podrobným formováním vztahů a interakcí mezi třídami a objekty se dostaneme k realizaci potřebného chování. Nejsnadnějším způsobem se tohoto cíle dobereme využitím diagramů aktivit a interakce (v této práci konkrétně využitého sekvenčního diagramu).

### 3.5.1 Sekvenční diagramy

Při modelování těchto diagramů interakce bylo čerpáno ze skutečných případů užití a převeden scénář UC do dynamické podoby posloupné komunikace mezi objekty za pomoci sekvenčního diagramu. Konkrétně na obrázku 3.3 byl zpracován UC1 (Evidence zaměstnanců).

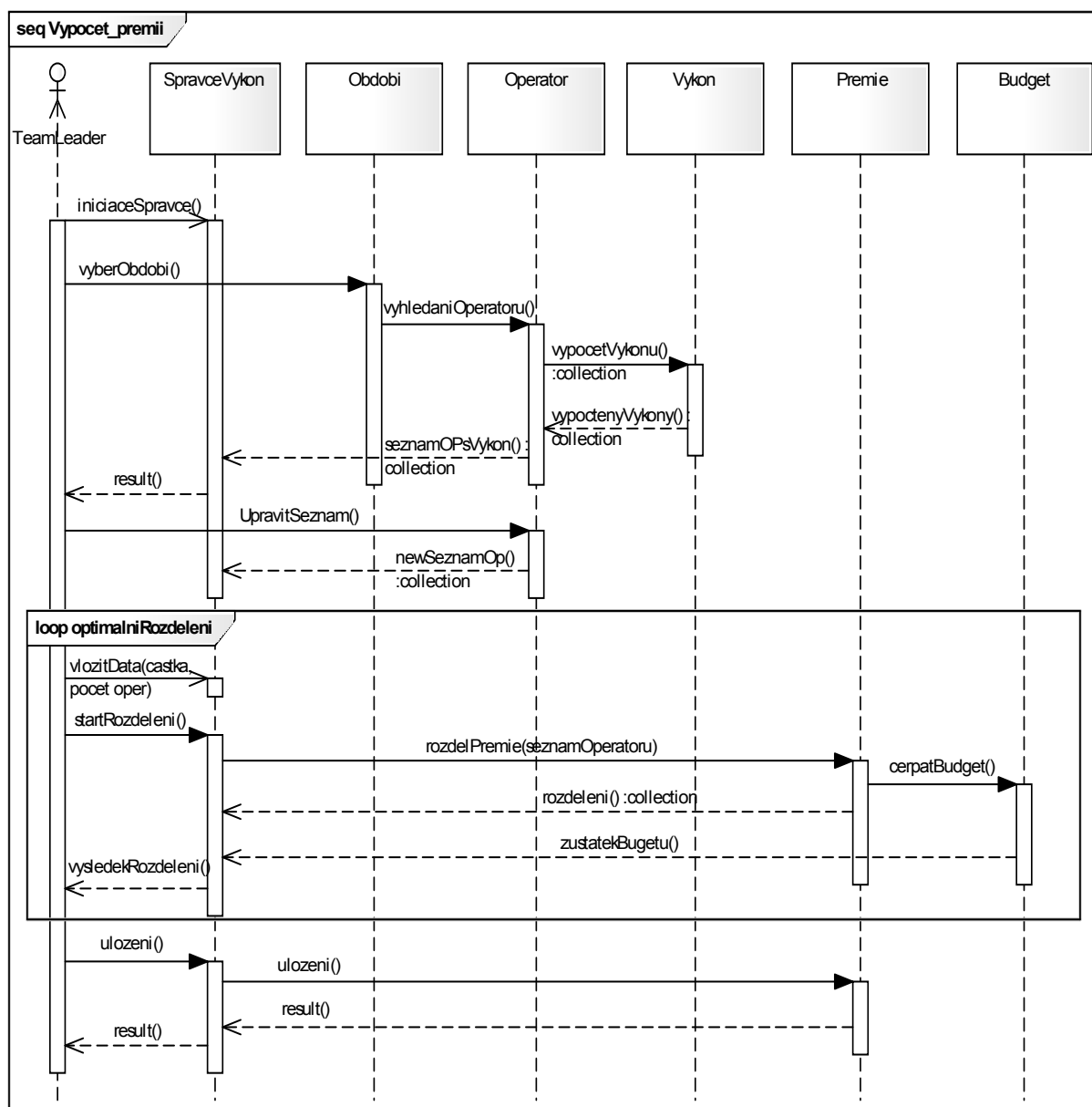


Obr. 3.3 Sekvenční diagram – evidence zaměstnanců.



Obr. 3.4 Sekvenční diagram – Zadání dat pro výkony.

Další sekvenční diagram popisující UC 8 (zadání dat pro výkony) je znázorněn na obrázku č. 3.4. Při realizaci scénáře byl vytvořen diagram vyobrazující interakci systému a TL, který vkládá důležitá data pro výpočet.

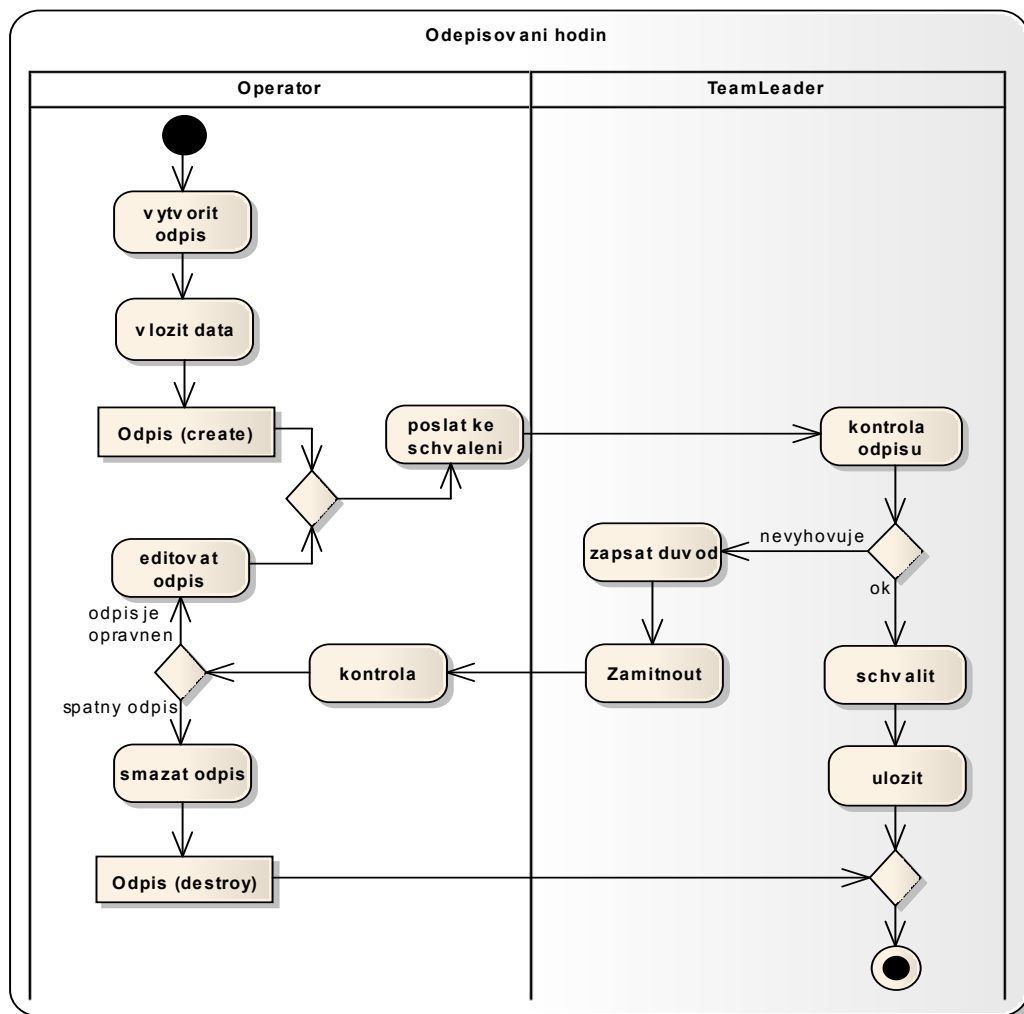


Obr. 3.5 Sekvenční diagram – výpočet výkonů.

Obrázek 3.5 zachycuje komunikaci mezi objekty aplikace při zpracování UC9 (výpočet prémie). Jednotlivé kroky výše vyobrazených diagramů odpovídají řádkům vybraného scénáře, a tudíž je zřejmý význam diagramu a není nutné tento proces znovu popisovat.

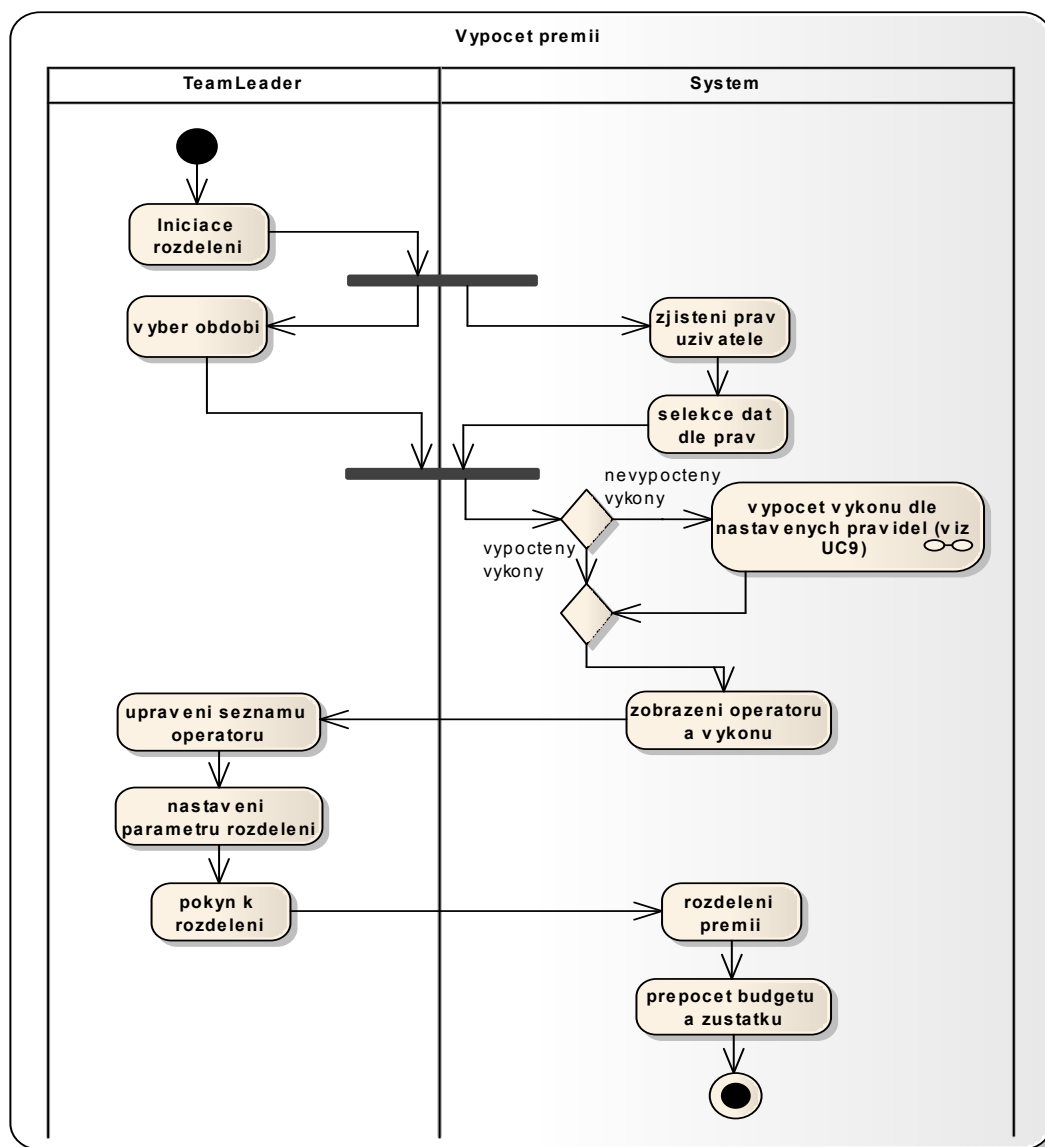
### 3.5.2 Diagramy aktivit

Pomocí tohoto nástroje lze zachytit také ostatní dynamické aspekty než jen přechod stavů a zasílání zpráv mezi objekty. Pro větší názornost jsou akce a aktivity na následujících obrázcích rozděleny do oddílů specifikujících účast jednotlivých účastníků systému nebo přímo vykonávající systém.



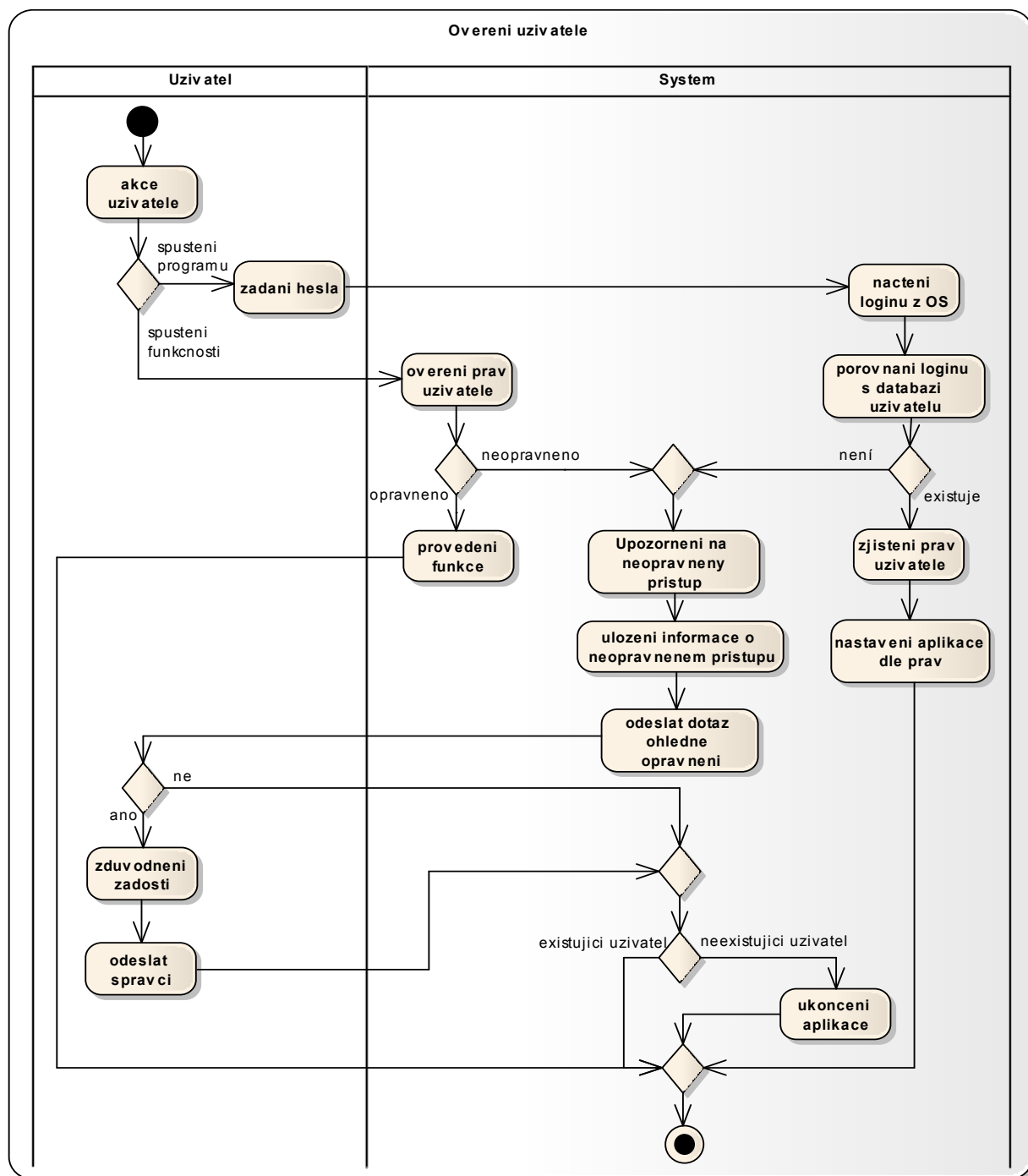
Obr. 3.6 Diagram aktivit – odepisování hodin.

Obrázek 3.6 znázorňuje proces, kterým projde každý záznam odpisu včetně větvení podle stanovených podmínek. Proces byl vytvořen kombinací dvou scénářů – UC6 (evidence záznamů odpisů) a UC7 (schválení odpisů). Nejdříve do procesu vstupuje operátor zadávající záznam odpisu a posílající odpis ke schválení. Poté sem vstupuje TL schvalující daný záznam. Při zamítnutí musí uvést důvod tohoto počínání, které se zobrazí operátoru. Ten následně zhodnotí důvod zamítnutí a rozhodne se, zda odpis smaže nebo jej upraví a pošle zpět ke schválení. Po schválení se odpis počítá do výkonu operátora a znepřístupní se jeho editace.



Obr. 3.7 Diagram aktivit – výpočet prémie.

Obrázek 3.7 zachycuje proces výpočtu prémie korespondující s UC9 (výpočet prémie), který probíhá v interakci mezi TL a zpracovávajícím systémem. TL iniciuje proces, nastavuje parametry výpočtu a zadává pokyn k přerozdělení prémie. Systém kontroluje práva uživatele, vybírá a třídí data, provádí výpočet výkonů a prémie, zobrazuje výsledky. Při nevhodném nastavení parametrů lze tento proces opakovat až do ukončení doby vybraného období (až do účetní uzávěrky).



Obr. 3.8 Diagram aktivit – ověření práv uživatele.

Bezpečnost aplikace je podchycena v případě užití UC18 (ověření práv uživatele) a znázorněna na obrázku 3.8. Prvního aktéra definuje obecný název „uživatel“, protože se může jednat o kteréhokoliv účastníka systému. Uživatel přistupující k aplikaci provádí určitou akci vyžadující zabezpečený přístup, která spouští ověřovací algoritmus systému. Pokud se daný uživatel nevyskytuje v databázi, nebo je jeho oprávnění nedostačující, systém upozorní uživatele na toto provinění a nabídne mu možnost požádat si o oprávnění. Na konci algoritmu systém zobrazí pouze informace dostupné k oprávnění uživatele.

### 3.6 Vytvoření prototypu

Dle předchozího návrhu a ve spolupráci se zaměstnanci společnosti byl vytvořen prototyp aplikace splňující téměř všechny specifikované požadavky. Některé funkcionality nebylo možné realizovat z důvodu omezených možností použitého softwaru a také kvůli bezpečnostním restrikcím společnosti. Jedná se například o import všech potřebných dat a export vypočtených premií ke zpracování účtárně.

K realizaci prototypu byla použita aplikace MS Access z kancelářského balíku MS Office. MS Access umožňuje tvorbu jednoduchého softwaru počínaje datovou základnou, přes aplikační vrstvu, až po vrstvu prezentační. Navíc nevyžaduje vysoké nároky na hardware, čímž zjednodušuje provoz aplikace, a za použití Visual Basicu rozšiřuje hranice své aplikovatelnosti. Jelikož společnost disponuje výše popsáním softwarem a realizace prototypu spadá do náplně této práce, prototyp nepřináší žádné finanční náklady.

Jelikož MS Access patří do skupiny relačních databází a předchozí návrh byl vytvořen z objektového pohledu, byl proveden proces mapování objektů do relační struktury. Jedná se o proces, při kterém nalezením optimální datové struktury umožňujeme objektům persistenci a mapujeme jejich vztahy. Při vytváření a optimalizaci relační databázové struktury nesmí být opomenuta normalizace dat, při níž dochází k odstranění anomálií z datového modelu.

Ve fázi návrhu prezentační vrstvy byl ve spolupráci se zaměstnanci společnosti vytvořen požadovaný vzhled prototypu (v rámci možností softwaru), jehož podobu je možné shlédnout v příloze č. 2, kde jsou uvedeny formuláře aktuálního (prototypového) softwarového řešení.

### 3.7 Návrh budoucí softwarové realizace

Pro realizaci kompletně navrženého systému je nutné zvolit jiný software umožňující implementaci všech požadovaných funkcionalit systému, poskytující vlastnosti a nástroje vyhovující bezpečnostní politice společnosti a disponující příjemnější vizualizací.

Pro synchronizaci se softwarovou strategií společnosti navrhuji využití aplikačních nástrojů implementovaných, podporovaných a vyvíjených ve společnosti. Pro datovou základnu navrhuji použití databáze Oracle. Pro aplikační a prezentační část navrhuji implementaci programovacího jazyka Java. Tento nástroj nabízí robustní podporu pro vývoj a implementaci dynamicky rozšiřitelných a různorodých systémů. Je podporován množstvím CASE nástrojů, které zjednodušují přechod mezi fázemi vývoje softwaru.



### 3.8 Předběžný časový plán projektu

Aby byl zajištěn úspěšný průběh návrhu systému, včetně všech fází vývoje, je nutné sestavit harmonogram činností, jejichž včasné a úspěšné dokončení povede k nadefinovaným cílům projektu. Všechny činnosti mají svoji dobu trvání, jsou prováděny v určitém pořadí s jasně danými návaznostmi a prováděny odpovědnými účastníky.

V následující tabulce 3.7 jsou vypsané jednotlivé činnosti směřující k vytvoření aplikace a jejímu nasazení. Doba trvání se počítá ve dnech a zkratka MD znamená ManDay (doslovný překlad zní „člověko-den“ a při výpočtu se 1 MD rovná 1 den práce zaměstnance na klasický pracovní úvazek, což se rovná 8 hodin čisté pracovní doby).

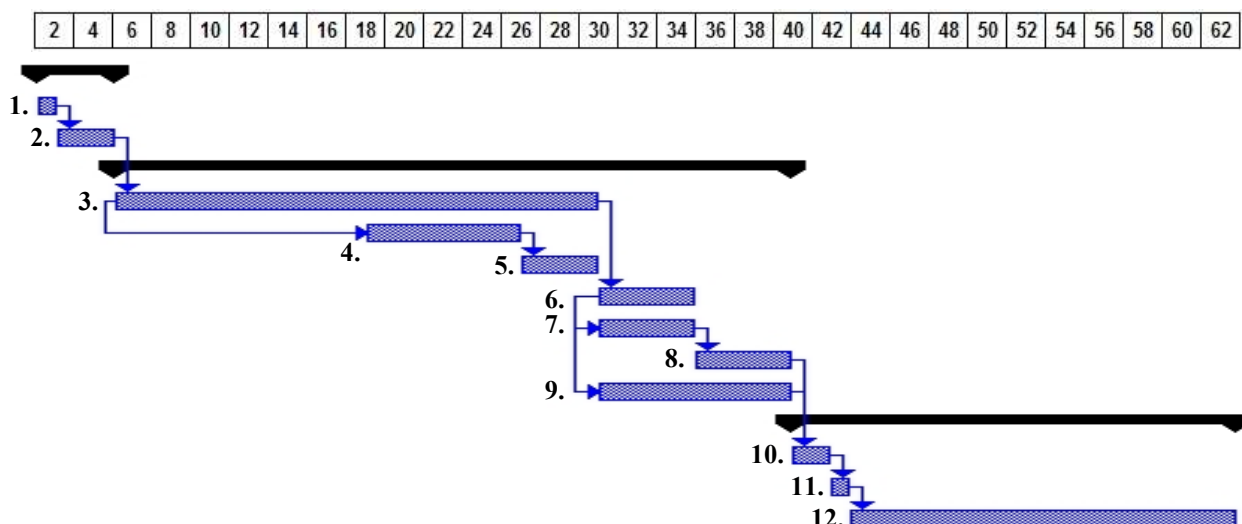
	Činnost	Doba trvání	MD	Účastníci
	<b>Návrh</b>			
1	Předložení návrhu k odsouhlasení	1	2	Analytik, Manager, TL
2	Rozpracování návrhu dle připomínek	3	3	Analytik
	<b>Realizace</b>			
3	Vytvoření aplikace	27	20	Programátor
4	Konzultace při tvorbě aplikace	8	4	Analytik, Programátor
5	Grafický návrh a konzultace	4	4	Analytik, Programátor, Uživatel
6	Alfa testování	5	6	Analytik, Programátor, Uživatel
7	Opravy chyb	5	3	Programátor
8	Beta testování	5	4	Analytik, Programátor, Uživatel
9	Tvorba dokumentace	10	8	Analytik, Programátor
	<b>Implementace</b>			
10	Školení uživatelů	2	4	Analytik, Uživatel
11	Nahrání iniciačních ostrých dat	1	1	Analytik, Uživatel
12	Zkušební provoz	20	10	Analytik, Uživatel
		<b>91</b>	<b>69</b>	

Tab. 3.7 Seznam činností.

Pojem uživatel v tabulce výše zahrnuje všechny dotčené uživatele systému (Manager, TL, Operátor). Činnosti byly rozděleny do tří hlavních skupin (návrh, realizace, implementace) seskupující logicky navazující činnosti do jednotlivých fází projektu. Popis činností:

1. **Předložení návrhu k odsouhlasení** – návrh vytvořený v této práci bude předložen ke schválení uživatelům, pro něž je aplikace určena.
2. **Rozpracování návrhu dle připomínek** – při schůzce, na které bude probíhat odsouhlasení návrhu, se mohou vyskytnout nápady, připomínky a zlepšení, které by měly být zapracovány do návrhu.

3. **Vytvoření aplikace** – v rámci tohoto úkolu bude připravený návrh včetně všech potřebných informací předán programátorovi, který započne s vývojem.
4. **Konzultace při tvorbě aplikace** – při vytváření aplikace si musí obě strany najít čas na vzájemné konzultace a ujasnění informací.
5. **Grafický návrh a konzultace** – pro design aplikace již byly stanoveny požadavky, ale výsledná podoba musí být navržena a konzultována s uživateli, aby byl výsledný vzhled aplikace optimální pro uživatele (příjemné barvy, rozmístění prvků atd.).
6. **Alfa testování** – odhalování chyb programu ve spolupráci uživatele, analytika a programátora. Většinou během prvního testování dojde k odhalení těch nejzávažnějších chyb ohrožujících funkcionality programu. Proto musí být vyhrazen čas pro následující činnost – opravu chyb.
7. **Opravy chyb** – programátor opravuje chyby zjištěné během testování a implementuje navržené změny.
8. **Beta testování** – při druhé fázi testování se ve většině případů vyskytují pouze drobné chyby, které programátor okamžitě opravuje.
9. **Tvorba dokumentace** – velmi důležitá činnost, během níž budou popsány všechny části aplikace, a to z několika pohledů (uživatelská dokumentace, dokumentace pro správce, technická dokumentace – popis programového kódu).
10. **Školení uživatelů** – nově vzniklý systém je nutné představit všem budoucím uživatelům a zaškolit je v používání nového softwaru.
11. **Nahrání iniciačních ostrých dat** – před uvedením do provozu musí být nahrána potřebná data a nastaveny všechny důležité parametry.
12. **Zkušební provoz** – i když aplikace prošla dvěma fázemi testování, může se projevit chyba až během práce s ostrými daty. Proto je nutné nejdříve pracovat ve zkušebním provozu při paralelním využívání nového i starého řešení.



Obr. 3.9 Ganttův diagram úkolů.

Ganttův diagram na obrázku 3.9 zobrazuje posloupnost všech činností a jejich vzájemné návaznosti. Některé činnosti nemohou být započaty dříve, než skončí předešlá činnost, a některé mohou probíhat současně. Součet doby trvání všech činností se rovná 91 dnům, ale při paralelním průběhu některých činností (viz Ganttův diagram) se celková doba trvání zkrátí na 64 dnů.

Doba trvání a počet MD se nemusí rovnat stejnému číslu z důvodu většího počtu účastníků provádějících danou činnost (případ, kdy MD převyšuje dobu trvání) nebo z důvodu nepřirazení celého časového fondu účastníka vybrané činnosti (případ kdy doba trvání převyšuje počet MD).

Potřebný počet MD pro úspěšné zajištění celého projektu byl ve spolupráci se zaměstnanci společnosti odhadnut na číslo **69**. Společnost disponuje nezbytným softwarem pro realizaci aplikace a žádné další náklady projekt nepřináší, a proto se celkový počet MD rovná výsledným nákladům na projekt.

## 4 Zhodnocení efektivity navrhovaného řešení

Jelikož společnost disponuje potřebnými složkami pro vytvoření navrhovaného systému (hardwarem i softwarem), budou všechny náklady vyjádřeny v podobě práce zaměstnanců měřených v jednotkách MD, o nichž byla zmínka v předešlé kapitole. Jelikož je aplikace vytvářena hlavně z důvodu úspory práce zaměstnanců, budou rovněž přínosy prezentovány v jednotkách MD.

Náklady a přínosy na zaměstnance nemohou být v této práci vypočteny finančně, protože by muselo být kalkulováno s platy zaměstnanců společnosti, jejichž zveřejnění, ať už v jakékoli podobě, se neslučuje se strategií a politikou společnosti.

Poněvadž se jedná o „jednoduchou“ evidenční aplikaci, klade management oddělení důraz na rychlou návratnost navrhovaného řešení. Tato návratnost byla stanovena na dva roky, kdy již v druhém roce by měly přínosy přesahovat náklady a v třetím roce by měly přínosy výrazně převyšovat drobné náklady údržby.

### 4.1 Očekávané náklady

#### 4.1.1 Jednorázové náklady

- Vytvoření návrhu – tato část byla připravena v rámci diplomové práce bez nároku na honorář, a tudíž se náklady návrhu rovnají **0 MD**.
- Příprava prototypu – prototyp byl realizován v rámci pracovní doby analytika. Při tvorbě a implementaci aplikace spolupracoval se všemi budoucími uživateli (Manager, TL, operátor). Celková pracnost všech zúčastněných aktérů byla vyčíslena na **15 MD**.
- Práce na projektu – z konceptu projektu popsaného v předešlé kapitole vyplývá celková pracnost na projektu na **69 MD**. Práce každého účastníka projektu byla konzultována přímo s danými pracovníky a s odborníky z IT oddělení společnosti, kteří mají s touto problematikou dlouholeté zkušenosti. Při rozpadu pracnosti na jednotlivé účastníky se analytik podílí 15 MD, programátor 35 MD a ostatní (uživatelé aplikace) 19 MD. Popis jednotlivých činností projektu se nachází v předešlé kapitole.
- **Celkové jednorázové náklady = 84 MD**

### 4.1.2 Provozní náklady

- Zálohování – zálohu provádí serverové automatizované nástroje. Proto není zapotřebí ručního zásahu a pracnost pro tuto činnost se rovná **0 MD**.
- Instalace – jelikož by se jednalo o webové intranetové řešení, nemusí probíhat softwarová správa spojená s instalací, upgradem a updatem aplikace. Proto náročnost této činnosti byla určena na **0 MD**.
- Školení – počítáme s přirozenou fluktuací pracovníků na oddělení, a tudíž je nutné počítat s pravidelným školením používání aplikace. Toto školení bude probíhat třikrát ročně pro zhruba pět uživatelů. Odhadovaná roční pracnost pro tuto činnost tvoří **4 MD/rok**.
- Údržba – při odhadu pracnosti údržby bylo předpokládáno s dobře vypracovaným návrhem zahrnujícím všechny potřebné funkcionality, s realizací všech těchto funkcionalit, dokonale provedené testovací fázi a odladění všech chyb a nedostatků systému. Pro tyto předpoklady a z důvodu možných drobných úprav byla pracnost údržby stanovena na **5 MD/rok**.
- **Celkové roční provozní náklady = 9 MD/rok**

## 4.2 Přínosy

Z předešlých částí práce vyplývá hlavní nesporný přínos aplikace, jenž bude spočívat v efektivitě práce s daty, kdy zautomatizováním skupiny činností a zjednodušením práce s daty se výrazně zdynamizuje proces výpočtu výkonů a prémie. Další přínos spočívá v nasazení nového sofistikovaného vzorce výpočtu, který umožňuje vyšší komplexnost a flexibilitu. Zefektivnění práce lze rozdělit do následujících činností:

- **Běžný výpočet výkonů** – jedná se o proces probíhající jednou měsíčně, který provádí zároveň všichni vedoucí týmů. Náplní procesu výpočtu výkonů je evidence potřebných dat, výpočet výkonů a přerozdělení prémie. Odhad úspor byl vytvořen ve spolupráci analytika s vedoucími týmů. Byly využity metody pozorování a měření, kdy přeměřením současného řešení a odečtu činností, které budou v aplikaci automatizovány, byl vypočten přínos nového řešení činící 4 hodiny na osobu a měsíc. Během prvního roku bude aplikace nasazena pro 4 týmy a v dalších letech pro další 4. Při výpočtu MD kalkulujeme s jednoduchým vzorcem, kdy se 1 MD rovná 8 hodinám (pracovní doba jednoho člověka na standardní úvazek).

- Přínos první rok:  $4\text{lidé} \cdot 4\text{hod} \cdot 12\text{měsíců} = 192\text{hod} / 8\text{prac.doba} = \mathbf{24\ MD/rok}$
- Přínos další roky:  $8\text{lídí} \cdot 4\text{hod} \cdot 12\text{měsíců} = 384\text{hod} / 8\text{prac.doba} = \mathbf{48\ MD/rok}$
- **Kontrola dat** – pravidelná činnost ověřující relevantnost informací a vypočtených výkonů. Navrhované řešení tuto činnost automatizuje a již ji není nutné manuálně zpracovávat. Proběhlo měření náročnosti dřívější činnosti, jejímž odebráním se tento čas rovná přínosu.
  - Přínos první rok:  $4\text{lidé} \cdot 3\text{hod} \cdot 12\text{měsíců} = 144\text{hod} / 8\text{prac.doba} = \mathbf{18\ MD/rok}$
  - Přínos další roky:  $8\text{lídí} \cdot 3\text{hod} \cdot 12\text{měsíců} = 288\text{hod} / 8\text{prac.doba} = \mathbf{36\ MD/rok}$
- **Výpočet a kontrola budgetů** – pravidelná činnost zabezpečující výpočet a správu budgetů. Nově by výpočty měly probíhat automatizovaně, pouze je zapotřebí úprava aktuálních podmínek, která zabírá minimální čas. Opět proběhlo měření této činnosti v současném procesu (vedoucí týmů a analytik) a dále odhadnut čas v novém automatizovaném procesu. Přínos se tedy rovná rozdílu starého a nového řešení.
  - Přínos první rok:  $4\text{lidé} \cdot 1\text{hod} \cdot 12\text{měsíců} = 48\text{hod} / 8\text{prac.doba} = \mathbf{6\ MD/rok}$
  - Přínos další roky:  $8\text{lídí} \cdot 1\text{hod} \cdot 12\text{měsíců} = 96\text{hod} / 8\text{prac.doba} = \mathbf{12\ MD/rok}$
- **Evidence a kontrola odpisů** – proces evidence v interakci mezi operátorem a vedoucím. Nové řešení výrazně zrychluje, zpřehledňuje a zefektivňuje práci s odpisy. Proces byl sledován a měřen ve spolupráci operátorů a analytika. Konečný odhad úspor provedl analytik s vedoucími týmů.
  - Přínos první rok:  $4\text{lidé} \cdot 3\text{hod} \cdot 12\text{měsíců} = 144\text{hod} / 8\text{prac.doba} = \mathbf{18\ MD/rok}$
  - Přínos další roky:  $8\text{lídí} \cdot 3\text{hod} \cdot 12\text{měsíců} = 288\text{hod} / 8\text{prac.doba} = \mathbf{36\ MD/rok}$
- **Reporting** – ke konci každého období jsou vytvářeny celkové výkonnostní přehledy po jednotlivých týmech a za celé oddělení. Navrhovaná aplikace by měla mít většinu potřebných přehledů již zakomponovánu a umožňovat rychlou tvorbu dalších přehledů. Úspory odhadnuty s tvůrci pravidelných přehledů.
  - Přínos první rok:  $4\text{lidé} \cdot 1\text{hod} \cdot 12\text{měsíců} = 48\text{hod} / 8\text{prac.doba} = \mathbf{6\ MD/rok}$
  - Přínos další roky:  $8\text{lídí} \cdot 1\text{hod} \cdot 12\text{měsíců} = 96\text{hod} / 8\text{prac.doba} = \mathbf{12\ MD/rok}$
- **Celkový přínos první rok: 72 MD/rok**
- **Celkový přínos další roky: 144 MD/rok**

### 4.3 Zhodnocení efektivity

Jednotky MD, představující odpracovanou lidskou práci v hodinách, jsou jiným vyjádřením vynaložených prostředků společnosti. Dostatečně odrážejí náklady i přínosy jednotlivých činností zaměstnance nebo celých projektů.

<b>Položky / Rok</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Náklady</b>			
Jednorázové	84		
Provozní	9	9	9
<b>Celkové</b>	<b>93</b>	<b>9</b>	<b>9</b>
<b>Přínosy</b>	72	144	144
<b>Efektivnost ročně (MD)</b>	<b>-21</b>	<b>135</b>	<b>135</b>
<b>Efektivnost kumulativně</b>	<b>-21</b>	<b>114</b>	<b>249</b>

Tab. 3.8 Výpočet efektivity.

Tabulka 3.8 obsahuje výpočet efektivity, z něhož vyplývá návratnost vytvoření aplikace již v druhém roce po implementaci (konkrétně v třetím měsíci druhého roku přínosy převyšují celkové náklady). Z výpočtu lze rovněž vyčíst evidentní přesah přínosu na konci druhého a třetího roku užívání aplikace.

Převodem jednotek MD do finančního vyjádření by vyčíslená efektivnost zvýšila svoji váhu a odrážela skutečné náklady a přínosy, protože každá společnost má nastavenou jinou výši platové politiky. Finanční vyjádření je nezbytné očistit o diskontní sazbu upravující hodnotu budoucích přínosů do podoby reálné současné hodnoty. Finanční vyčíslení 1 MD nebo dokonce platy lidí ve společnosti GE Money není možné v této práci uvést z důvodu strategie a politiky společnosti. Obecné vyjádření MD pomocí průměrných mezd by dostatečně nevypovídalo o skutečných přínosech navrhovaného projektu.

## Závěr

Cílem diplomové práce bylo navrhnout aplikaci spravující komplexní prémiový model oddělení Akvizic společnosti GE Money Česká republika, jejíž hlavním účelem je zefektivnění práce uživatelů a zřehlednění celého modelu.

Při návrhu aplikace jsem spolupracoval s pracovníky společnosti a vycházel jsem ze strategie společnosti, strategie oddělení a z požadavků pracovníků oddělení. Rovněž jsem při návrhu musel zohlednit tvorbu prototypu a návratnost tvorby aplikace do dvou let, což byly požadavky vedení oddělení Akvizic.

V první části této práce rozebírám teoretická východiska, kde popisují vlastnosti všech použitých nástrojů a jejich možné použití. Před tvorbou návrhu jsem konzultoval použití UML s programátory a vybral si pouze ty diagramy, které podle mého názoru dostatečně popisují vybranou problematiku a nabízejí možnost zapracování všech funkčních i nefunkčních požadavků.

Druhá kapitola je věnována popisu problémové oblasti a analýze současného stavu řešené problémové oblasti.

Návrhem aplikace se zabírám ve třetí kapitole. Nejdříve se zmiňuji o optimalizaci procesu a vytvoření nového algoritmu výpočtu, protože tyto činnosti výrazně ovlivňují správnou funkčnost budoucí aplikace. Dále byly specifikovány požadavky, vytvořeny případy užití, včetně scénářů, analyzovány případy užití a vytvořen analytický model tříd popisující strukturu systému. Pro zobrazení dynamického chování systému byly vymodelovány sekvenční a aktivity diagramy.

Součástí práce je také zmínka o tvorbě prototypu a navrhované budoucí softwarové realizaci nové aplikace. Prototyp měl původně sloužit jen pro simulaci procesu a kontrolu nového algoritmu výpočtu výkonů a premií. Jelikož simulace dopadla úspěšně, byly v aplikaci postupně implementovány další a další funkcionality, které pokrývají téměř všechny požadavky specifikované v návrhu (kromě požadavků, které z technických důvodů použitého řešení není možné realizovat). Pro velký úspěch prototypu z hlediska funkčnosti a přehlednosti byla odsouhlasena jeho implementace a spuštěno dočasné využívání až do doby realizace navrhovaného softwarového řešení.



Důležitou součástí je také předběžný plán projektu, znázorňující posloupnost činností vedoucí k dosažení úspěšného nasazení aplikace. Během této části byla odhalena celková doba projektu činící 64 dní.

Zhodnocení efektivnosti navrhovaného řešení dokumentuje čtvrtá kapitola. Při výpočtu nákladů a přínosů bylo kalkulováno pouze s prací zaměstnanců, protože jiné náklady a přínosy aplikace zatím nebyly odhaleny. Při výpočtu efektivnosti byla zjištěna návratnost již ve třetím měsíci druhého roku po nasazení aplikace a také propočteny přínosy na konci druhého a třetího roku, jež dokazují významný přínos aplikace.

Z předchozího textu vyplývá úspěšná implementace všech požadavků a splnění všech podmínek návrhu. Z tohoto hlediska si dovoluji usuzovat, že bylo dosaženo cíle mé diplomové práce.

## Seznam použité literatury

- [1] ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací: Objektově orientovaná analýza a návrh prakticky*. 2. vyd. Brno: Computer Press, 2007. 567 s. ISBN 978-80-251-1503-9
- [2] FOWLER, M. *UML bez záhad*. 3.vyd. Praha: GRADA Publishing, a.s., 2007. 312 s. ISBN 80-247-2062-0
- [3] GRASSEOVÁ, M. *Procesní řízení: Ve veřejném i soukromém sektoru*. 2. vyd. Brno: Computer Press, 2008. 272 s. ISBN 978-80-251-1987-7
- [4] KALUŽA, J. *Tvorba datového modelu v prostředí strategických informačních systémů*. 1. vyd. Ostrava: Grafie, 1996. 115 s.
- [5] KANISOVÁ, H., MÜLLER, M. *UML srozumitelně*. 2. vyd. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4
- [6] MCCONNELL, S. *Odhadování softwarových projektů: Jak správně určit rozpočet, termíny a zdroje*. 1. vyd. Brno: Computer Press, 2006. 320 s. ISBN 80-251-1240-3
- [7] SCHWALBE, K. *Řízení projektů v IT: Kompletní průvodce*. 2.vyd. Brno: Computer Press, 2007. 720 s. ISBN 978-80-251-1526-8
- [8] WIEGERS, K. E. *Požadavky na software: Od zadání k architektuře aplikace*. 2.vyd. Brno: Computer Press, 2008. 448 s. ISBN 978-80-251-1877-1
- [9] GE CZ, *Internetové stránky společnosti GE Česká a Slovenská republika* [cit. 2009-04-01]. Dostupný z WWW: <http://www.ge.com/cz/>
- [10] GE MONEY CZ, *Internetové stránky společnosti GE Money Česká republika* [cit. 2009-04-01]. Dostupný z WWW: [www.gemoney.cz](http://www.gemoney.cz)

## **Seznam zkratek**

UML – Unified Modeling Language

CASE – Computer-Aided Software Engineering

OOP – objektově orientované programování

UC – Use Case diagram

RTM – Requirements Traceability Matrix

TL – Team leader

MNG - Manager

MD – Man/Day

PV – Present Value

FV– Future Value

## Seznam tabulek

Tab. 1.1 Rozdělení diagramů.

Tab. 1.2 Matice RTM.

Tab. 1.3 Definice ornamentů viditelnosti.

Tab. 1.4 Typy asociací.

Tab. 3.1 Seznam požadavků.

Tab. 3.2 Definice priorit.

Tab. 3.3 Seznam případů užití.

Tab. 3.4 RTM matice sledovanosti požadavků.

Tab. 3.5 Specifikace případu užití – evidence zaměstnanců.

Tab. 3.6 Kandidátka tříd, atributů a odpovědností.

Tab. 3.7 Seznam činností.

Tab. 3.8 Výpočet efektivnosti.

## Seznam obrázků

Obr. 1.1 Struktura UML.

Obr. 1.2 Komponenty UC diagramů.

Obr. 1.3 Notace tříd.

Obr. 1.4 Notace tříd.

Obr. 1.5 Diagram aktivit.

Obr. 1.6 Podrobnější diagram aktivit.

Obr. 1.7 Sekvenční diagram.

Obr. 1.8 Komunikační diagram.

Obr. 3.1 Diagram případu užití.

Obr. 3.2 Diagram analytických tříd.

Obr. 3.3 Sekvenční diagram – evidence zaměstnanců.

Obr. 3.4 Sekvenční diagram – Zadání dat pro výkony.

Obr. 3.5 Sekvenční diagram – výpočet výkonů.

Obr. 3.6 Diagram aktivit – odepisování hodin.

Obr. 3.7 Diagram aktivit – výpočet prémie.

Obr. 3.8 Diagram aktivit – ověření práv uživatele.

Obr. 3.9 Ganttův diagram úkolů.

## **Seznam Příloh**

Příloha č. 1 – Specifikace případu užití

Příloha č. 2 – Formuláře prototypu

Příloha č. 3 – Informační popis prototypu

## Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo,
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3),
- souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO,
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona,
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 28.4.2009

.....  
Miroslav Hon

Adresa trvalého pobytu studenta:

Skřipov 188, 747 45 Skřipov

## Přílohy

### Příloha č. 1 – Specifikace případu užití

<b>Případ užití: Evidence zaměstnanců</b>
<b>ID: UC1</b>
<b>Primární aktéři:</b> Team leader Manager <b>Vedlejší aktéři:</b> Operátor
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Operátor poskytne osobní data.
<b>Hlavní scénář:</b> 1. Seznam zaměstnanců je spuštěn primárním aktérem. 2. Když PA přidává záznam: 2.1 PA iniciuje vytvoření záznamu. 2.2 PA vkládá data. 3. Když PA edituje záznam: 3.1 PA vyhledá záznam. 3.2 PA edituje záznam. 4. Primární aktér zadá pokyn k uložení záznamu. 5. Záznam zaměstnance je uložen do databáze.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje evidenci. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží záznam a zobrazí informaci o uložení. 2. Změny v evidenci se projeví v celém systému. (PA- primární aktér)

<b>Případ užití: Evidence pravidel</b>
<b>ID: UC2</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.
<b>Hlavní scénář:</b> 1. Evidence pravidel je spuštěna primárním aktérem. 2. Primární aktér přidává (edituje) údaje pravidla. 3. Primární aktér vloží data do systému. 4. Primární aktér zadá pokyn k uložení pravidel. 5. Záznam pravidla je uložen do databáze.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje evidenci. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží záznam a zobrazí informaci o uložení. 2. Změny v evidenci se projeví v celém systému.

<b>Případ užití: Evidence typů odpisů</b>
<b>ID: UC3</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.
<b>Hlavní scénář:</b> 1. Evidence typů odpisů je spuštěna primárním aktérem. 2. Primární aktér přidává (edituje) údaje typ odpisu. 3. Primární aktér vloží data do systému. 4. Primární aktér zadá pokyn k uložení typu odpisu. 5. Záznam typu odpisu je uložen do databáze.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje evidenci. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží záznam a zobrazí informaci o uložení. 2. Změny v evidenci se projeví v celém systému.

(PA- primární aktér)



<b>Případ užití: Evidence budgetů</b>
<b>ID: UC4</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.
<b>Hlavní scénář:</b> 1. Evidence budgetů je spuštěna primárním aktérem. 2. Primární aktér přidává (edituje) údaje budget. 3. Primární aktér vloží data do systému. 4. Primární aktér zadá pokyn k uložení budgetu. 5. Záznam pravidla je uložen do databáze.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje evidenci. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží záznam a zobrazí informaci o uložení. 2. Změny v evidenci se projeví v celém systému.

<b>Případ užití: Import dat</b>
<b>ID: UC5</b>
<b>Primární aktéři:</b> Team leader MIS
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Dostupná data v externích databázích (MIS). 4. Funkčnost ODBC rozhraní.
<b>Hlavní scénář:</b> 1. Spuštění importovacího nástroje primárním aktérem. 2. Primární aktér zadává kritéria importu. 3. Primární aktér dá pokyn k importu. 4. Systém importuje data dle zadaných kritérií.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje import dat. 3. Neuložení záznamu (nekorektní data, import historických dat).
<b>Následné podmínky:</b> 1. Systém uloží data a zobrazí informaci o importu. 2. Importovaná data se projeví ve výpočtech výkonů.

(PA- primární aktér)

<b>Případ užití: Evidence záznamů odpisů</b>
<b>ID: UC6</b>
<b>Primární aktéři:</b> Operátor
<b>Vstupní podmínky:</b> <ol style="list-style-type: none"> <li>1. Funkčnost aplikace (aplikační i serverové části).</li> <li>2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.</li> <li>3. Existence dat v číselníku typů odpisů.</li> <li>4. Vybraný záznam zatím není schválen nadřízeným.</li> </ol>
<b>Hlavní scénář:</b> <ol style="list-style-type: none"> <li>1. Spuštění formuláře záznamů odpisů primárním aktérem.</li> <li>2. Když se PA rozhraní nový záznam: <ol style="list-style-type: none"> <li>2.1 PA vyvolá příkaz pro přidání záznamu.</li> <li>2.2 PA vloží data k záznamu odpisu.</li> </ol> </li> <li>3. Když PA upravuje záznam: <ol style="list-style-type: none"> <li>3.1 PA vybírá záznam, který bude upravovat.</li> <li>3.2 PA vyvolá rozhraní pro úpravu záznamu.</li> <li>3.3 PA edituje data k záznamu odpisu.</li> </ol> </li> <li>4. Primární aktér dá pokyn k uložení.</li> <li>5. Systém ukládá data do databáze.</li> </ol>
<b>Alternativní Scénáře:</b> <ol style="list-style-type: none"> <li>1. Ukončení aplikace - chybový stav při porušení práv aplikace.</li> <li>2. Primární aktér stornuje zadání dat.</li> <li>3. Neuložení záznamu (nekorektní data, změna historických dat)</li> </ol>
<b>Následné podmínky:</b> <ol style="list-style-type: none"> <li>1. Systém uloží data a zobrazí informaci o uložení.</li> <li>2. Data jsou viditelná u Team leadera v seznamu záznamů ke schválení.</li> </ol>

(PA- primární aktér)

<b>Případ užití: Schválení odpisů</b>
<b>ID: UC7</b>
<b>Primární aktéři:</b> Team Leader
<b>Vstupní podmínky:</b> <ol style="list-style-type: none"> <li>1. Funkčnost aplikace (aplikační i serverové části).</li> <li>2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.</li> <li>3. Operátor vytvořil (upravil) záznam odpisu.</li> </ol>
<b>Hlavní scénář:</b> <ol style="list-style-type: none"> <li>1. Spuštění formuláře schvalování odpisů primárním aktérem.</li> <li>2. Systém vyhledá a zobrazí záznamy spadající do působnosti PA.</li> <li>3. Primární aktér vybírá záznam.</li> <li>4. Když PA schvaluje záznam: <ol style="list-style-type: none"> <li>4.1 PA klikne na tlačítko „Schválit“.</li> </ol> </li> <li>5. Když PA zamítá záznam: <ol style="list-style-type: none"> <li>5.1 PA vkládá důvod zamítnutí.</li> <li>5.2 PA potvrzuje zamítnutí tlačítkem zamítnout.</li> </ol> </li> <li>6. Systém ukládá záznam do databáze.</li> </ol>
<b>Alternativní Scénáře:</b> <ol style="list-style-type: none"> <li>1. Ukončení aplikace - chybový stav při porušení práv aplikace.</li> <li>2. Primární aktér stornuje zadání dat.</li> <li>3. Neuložení záznamu (nekorektní data, změna historických dat).</li> </ol>
<b>Následné podmínky:</b> <ol style="list-style-type: none"> <li>1. Systém uloží data a zobrazí informaci o uložení.</li> <li>2. Data jsou viditelná u Operátora v seznamu schválených odpisů.</li> </ol> <p>(PA- primární aktér)</p>

<b>Případ užití: Zadání dat pro výkony</b>
<b>ID: UC8</b>
<b>Primární aktéři:</b> Team Leader
<b>Vstupní podmínky:</b> <ol style="list-style-type: none"> <li>1. Funkčnost aplikace (aplikační i serverové části).</li> <li>2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.</li> <li>3. PA má všechny potřebné informace k zadání periodických dat týkající se výkonů.</li> </ol>
<b>Hlavní scénář:</b> <ol style="list-style-type: none"> <li>1. Spuštění rozhraní pro zadávání dat pro výkony primárním aktérem.</li> <li>2. PA vybírá období pro zadání dat.</li> <li>3. Systém vyhledá operátory spadající do působnosti PA.</li> <li>4. Pro všechny vybrané operátory: <ol style="list-style-type: none"> <li>4.1 Systém zobrazí ID a název operátora.</li> <li>4.2 PA vkládá data k tomuto záznamu.</li> </ol> </li> <li>5. Primární aktér zadá pokyn k uložení dat.</li> <li>6. Systém uloží data do databáze.</li> </ol>
<b>Alternativní Scénáře:</b> <ol style="list-style-type: none"> <li>1. Ukončení aplikace - chybový stav při porušení práv aplikace.</li> <li>2. Primární aktér stornuje zadání dat.</li> <li>3. Neuložení záznamu (nekorektní data, změna historických dat).</li> </ol>
<b>Následné podmínky:</b> <ol style="list-style-type: none"> <li>1. Systém uloží data a zobrazí informaci o uložení.</li> <li>2. Data jsou k dispozici pro výpočet výkonů.</li> </ol>

(PA- primární aktér)

<b>Případ užití: Výpočet prémie</b>
<b>ID:</b> UC9
<b>Primární aktéři:</b> Team Leader
<b>Vstupní podmínky:</b> <ol style="list-style-type: none"> <li>1. Funkčnost aplikace (aplikační i serverové části).</li> <li>2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.</li> <li>3. Existují data pro dané období: <ol style="list-style-type: none"> <li>3.1 Zadány budgety, pravidla a operátoři.</li> <li>3.2 Úspěšný import dat.</li> <li>3.3 Vloženy data pro výpočet výkonů.</li> </ol> </li> </ol>
<b>Hlavní scénář:</b> <ol style="list-style-type: none"> <li>1. Spuštění rozhraní pro výpočet prémie PA.</li> <li>2. PA vybírá období pro výpočet.</li> <li>3. Systém vyhledá operátory spadající do působnosti PA.</li> <li>4. Systém spočítá a zobrazí výkony operátorů.</li> <li>5. Pro všechny operátory nemající nárok na prémii: <ol style="list-style-type: none"> <li>5.1 PA vybere záznam operátora.</li> <li>5.2 PA nastaví atribut „nárok na prémii“ na FALSE.</li> </ol> </li> <li>6. Dokud PA není spokojen s rozdělením prémie: <ol style="list-style-type: none"> <li>6.1 PA vloží částku, kterou požaduje čerpat z budgetu.</li> <li>6.2 PA vloží počet operátorů, kterým budou rozděleny prémii.</li> <li>6.3 PA dá pokyn k rozdělení budgetu operátorům.</li> <li>6.4 Systém rozpočítá budget a rozdělí prémii dle zadaných podmínek.</li> </ol> </li> <li>7. PA dá pokyn k uložení dat.</li> <li>8. Systém uloží data do databáze.</li> </ol>
<b>Alternativní Scénáře:</b> <ol style="list-style-type: none"> <li>1. Ukončení aplikace - chybový stav při porušení práv aplikace.</li> <li>2. Primární aktér stornuje zadání dat.</li> <li>3. Neuložení záznamu (nekorektní data, změna historických dat).</li> <li>4. Přerozdělení prémie</li> </ol>
<b>Následné podmínky:</b> <ol style="list-style-type: none"> <li>1. Systém uloží data a zobrazí informaci o uložení.</li> <li>2. Data jsou viditelná v přehledech výkonů.</li> </ol>

(PA- primární aktér)

<b>Případ užití: Přerozdělení prémie</b>
<b>ID: UC10</b>
<b>Primární aktéři:</b> Team Leader
<b>Vstupní podmínky:</b> <ol style="list-style-type: none"> <li>1. Funkčnost aplikace (aplikační i serverové části).</li> <li>2. Aktér existuje v databázi uživatelů a má patřičné oprávnění.</li> <li>3. Existují data pro dané období: <ol style="list-style-type: none"> <li>3.1 Zadány budgety, pravidla a operátoři.</li> <li>3.2 Úspěšný import dat.</li> <li>3.3 Vloženy data pro výpočet výkonů.</li> <li>3.4 Úspěšně proveden UC9.</li> </ol> </li> </ol>
<b>Hlavní scénář:</b> <ol style="list-style-type: none"> <li>1. Spuštění rozhraní pro výpočet prémie PA.</li> <li>2. PA vybírá období pro výpočet.</li> <li>3. Systém vyhledá operátory spadající do působnosti PA.</li> <li>4. Systém zobrazí výkony a prémie operátorů.</li> <li>5. PA odemyká záznamy pro přerozdělení prémie.</li> <li>6. Dokud PA není spokojen s rozdělením prémie: <ol style="list-style-type: none"> <li>6.1 Pro operátory vybrané PA: <ol style="list-style-type: none"> <li>6.1.1 PA zvolí záznam operátora.</li> <li>6.1.2 PA zvolí položku (osobní ohodnocení, týmová složka, srážky).</li> <li>6.1.3 PA vkládá částku.</li> </ol> </li> <li>6.2 PA ukládá záznamy.</li> <li>6.3 Systém uloží data do databáze.</li> <li>6.4 Systém zobrazí aktuální zůstatek budgetu.</li> </ol> </li> </ol>
<b>Alternativní Scénáře:</b> <ol style="list-style-type: none"> <li>1. Ukončení aplikace - chybový stav při porušení práv aplikace.</li> <li>2. Primární aktér stornuje zadání dat.</li> <li>3. Neuložení záznamu (nekorektní data, změna historických dat).</li> </ol>
<b>Následné podmínky:</b> <ol style="list-style-type: none"> <li>1. Systém uloží data a zobrazí informaci o uložení.</li> <li>2. Data jsou viditelná v přehledech výkonů.</li> </ol>

(PA- primární aktér)

<b>Případ užití: Proplacení</b>
<b>ID: UC11</b>
<b>Primární aktéři:</b> Team Leader Účtárna
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data pro dané období. 4. Team leader již má přerozděleny prémie. 5. Nastalo období k proplácení prémie účtárnou.
<b>Hlavní scénář:</b> 1. Spuštění rozhraní pro výpočet prémie PA. 2. PA vybírá období pro výpočet. 3. Systém vyhledá operátory spadající do působnosti PA. 4. Systém zobrazí výkony a prémie operátorů. 5. PA zadá pokyn k proplacení. 6. Systém zobrazí seznam operátorů s částkami k proplacení a ověřuje zda PA opravdu miní proplatit tyto částky. 7. Když PA miní proplatit uvedené částky: 7.1 PA potvrdí proplacení. 7.2 Systém pošle data účtárně. 7.3 Systém zobrazí výsledek proplacení. 8. Když PA nesouhlasí s částkami: 8.1 PA stornuje proplacení. 8.2 PA se vrací k výpočtu a přerozdělení prémie (UC9, UC10)
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje zadání dat. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém uloží data a zobrazí informaci o proplacení. 2. Prémie jsou proplaceny operátorům.

(PA- primární aktér)

<b>Případ užití: Přehledy</b>
<b>ID: UC12</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy.
<b>Hlavní scénář:</b> 1. PA spouští přehledy. 2. Systém vyhodnocuje práva PA odfiltruje data spadající do působnosti PA. 3. Systém zobrazí hlavní reportovací dashboard s aktuálními přehledy (poslední období). 4. Dokud PA nezíská požadované informace: 4.1 PA zadává kritéria. 4.1 Systém zobrazí report za dané období.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace.
<b>Následné podmínky:</b> 1. Systém zobrazí požadovaný přehled.

<b>Případ užití: Přehledy budgetů</b>
<b>ID: UC13</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy.
<b>Hlavní scénář:</b> 1. PA spouští přehledy a vybírá podnabídku přehledy budgetů. 2. Systém vyhodnocuje práva PA odfiltruje data spadající do působnosti PA. 3. Systém zobrazí aktuální přehledy budgetů (poslední období). 4. Dokud PA nezíská požadované informace: 4.1 PA zadává kritéria. 4.1 Systém zobrazí report za dané období.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace.
<b>Následné podmínky:</b> 1. Systém zobrazí požadovaný přehled.

(PA- primární aktér)



<b>Případ užití: Přehledy odpisů</b>
<b>ID: UC14</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy.
<b>Hlavní scénář:</b> 1. PA spouští přehledy a vybírá podnabídku přehledy odpisů. 2. Systém vyhodnocuje práva PA odfiltruje data spadající do působnosti PA. 3. Systém zobrazí aktuální report odpisů (poslední období). 4. Dokud PA nezíská požadované informace: 4.1 PA zadává kritéria. 4.1 Systém zobrazí report za dané období.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace.
<b>Následné podmínky:</b> 1. Systém zobrazí požadovaný přehled.

<b>Případ užití: Přehledy výkonů</b>
<b>ID: UC15</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy.
<b>Hlavní scénář:</b> 1. PA spouští přehledy a vybírá podnabídku přehledy výkonů. 2. Systém vyhodnocuje práva PA odfiltruje data spadající do působnosti PA. 3. Systém zobrazí aktuální report výkonů (poslední období). 4. Dokud PA nezíská požadované informace: 4.1 PA zadává kritéria. 4.1 Systém zobrazí report za dané období.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace.
<b>Následné podmínky:</b> 1. Systém zobrazí požadovaný přehled.

(PA- primární aktér)

<b>Případ užití: Přehledy osobních výkonů</b>
<b>ID: UC16</b>
<b>Primární aktéři:</b> Operátor
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy.
<b>Hlavní scénář:</b> 1. PA spouští přehledy a systém jej automaticky přesměruje na podnabídku přehledy osobních výkonů. 2. Systém vyhodnocuje práva PA odfiltruje data spadající do působnosti PA. 3. Systém zobrazí aktuální report výkonů (poslední období). 4. Dokud PA nezíská požadované informace: 4.1 PA zadává kritéria. 4.1 Systém zobrazí report za dané období.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace.
<b>Následné podmínky:</b> 1. Systém zobrazí požadovaný přehled.

<b>Případ užití: Export přehledu</b>
<b>ID: UC17</b>
<b>Primární aktéři:</b> Team leader Manager
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Aktér existuje v databázi uživatelů a má patřičné oprávnění. 3. Existují data potřebná pro přehledy. 4. Je spuštěn přehled viz UC12 až UC15.
<b>Hlavní scénář:</b> 1. PA spouští nabídku export přehledu v UC12 až UC15. 2. PA vybírá kritéria exportu a dá pokyn k exportu. 3. Systém vyzve PA k zadání diskového umístění exportovaného souboru. 4. PA zadá umístění a dá pokyn k uložení exportu. 5. Systém exportuje přehled do formátu MS Excel.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje zadání dat. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Systém vytvoří soubor s exportovanými daty. (PA- primární aktér)

<b>Případ užití: Ověření práv uživatele</b>
<b>ID: UC18</b>
<b>Primární aktéři:</b> Team leader Manager Operátor
<b>Vstupní podmínky:</b> 1. Funkčnost aplikace (aplikační i serverové části). 2. Existence číselníku uživatelů.
<b>Hlavní scénář:</b> 1. PA spouští akci (aplikaci nebo určitý UC-funkcionalitu). 2. Když PA spouští aplikaci: 2.1 Systém zjistí login PA z operačního systému. 2.2 PA zadává heslo. 3. Systém porovnává login PA s číselníkem uživatelů. 4. Systém ověřuje práva PA. 5. Když PA nemá patřičné oprávnění: 5.1 Systém upozorní PA ohledně nedovoleného chování v aplikaci. 5.2 Systém nepustí uživatele dále (např. neotevře podmodul, formulář). 5.3 Systém zašle dotaz zda PA chce udělit patřičná oprávnění. 5.4. Když PA potvrdí žádost o oprávnění: 5.4.1 PA zdůvodní požadavek k udělení oprávnění. 5.4.2 PA odesílá požadavek správci. 6. Když PA neexistuje v číselníku: 6.1 Systém ukončí aplikaci. 7. Systém předává informace ohledně oprávnění PA.
<b>Alternativní Scénáře:</b> 1. Ukončení aplikace - chybový stav při porušení práv aplikace. 2. Primární aktér stornuje zadání dat. 3. Neuložení záznamu (nekorektní data, změna historických dat).
<b>Následné podmínky:</b> 1. Uživatel je upozorněn na nedostatečné oprávnění nebo se jeho informace zobrazí v záhlaví aplikace.

(PA- primární aktér)

## Příloha č. 2 – Formuláře prototypu

Pozn.: data uvedená v této práci jsou pouze testovací a nekorespondují s realitou. Podobnost se skutečností může být čistě náhodná. Data ve formulářích a sestavách jsou pouze testovací.

### Formulář F1A: Přidání a úprava operátora (uživatele)

F1A

Výkony a prémie 1.0

GE Money CZ

Operátor / uživatel

Uživatel: 678952478

Miroslav Hon

Proces: DDTeam

Dnešní datum: 8.3.2009

ID - SSO: 123040482

Jméno: Miroslav

Příjmení: Hon

Kód CC: HH22

Kód UFO: HM76KO

Proces: DDTeam

Práva: Team Leader

Úvazek: 40

Plat: 5 000

Platnost změny: 1.2.2009

Aktivní: ☒

Historie změn:

Datum změny	Aktor změny	Popis změn	Datum platnosti změny
11. 2.2009	Velitel Vedoucí	Vytvořeno.	1. 1.2008
5. 2.2009	Velitel Vedoucí	Změněno: plat, úvazek	1. 2.2009

Upravit SSO

Detail Změny

Uložit změny

Zavřít

### Formulář F2A: Evidence pravidel – váhy kvality a kvantity

F2A

Výkony a prémie 1.0

GE Money CZ

Váha kvality

Uživatel: 678952478

Miroslav Hon

Proces: DDTeam

Dnešní datum: 8.3.2009

Proces: SOM

Váha kvality: 0,8

Váha kvantity: 0,2

Platnost změny: 16.12.2007 - 15.1.2008

Historie změn:

Datum změny	Aktor změny	Období platnosti změny	Proces	Váha kvality	Váha kvantity
1. 1.2008	Mira	16.12.2007 - 15.1.2008	SOM	0,8	0,2

Uložit změny

Zavřít

## Formulář F6: Výpočet výkonů a rozdělení prémie

**F6 Výkony a prémie 1.0 - GE Money CZ**

### Výsledné výkony a prémie

Uživatel: 678952478 Miroslav Hon Proces: DDTeam Dnešní datum: 8.3.2009

Vyberte období: 16.8.2008 - 15.9.2008

Budget /běžná období: 15795 Čerpat: 7000

Zustatek /běžná období: 4465 Zůstatek /minulá období: 6217

Budget (rok) Emu, GE šek: 14400 Zůstatek /Emu, GE šek: 11900

Kolik lidem dát prémie: Výpočet prémie

Číslo prémie	SSO	Operátor/administrátor	Váha kvantity	Výkon	Váha kvality	Kvalita	Celkový výkon	Koef. prémie	Prémie za výkon	Prémie čistá	Týmová složka	Osobní složka	Srážky	Celkem prémie	GE šek	EMU
<input checked="" type="checkbox"/>	324567875	Novák Miroslav	75,0%	29,8%	25,0%	96,00	46,4%	38,1%	1848	1052	0	0	-500	552	0	0
<input checked="" type="checkbox"/>	988765345	Niklová Lenka	75,0%	32,8%	25,0%	80,00	44,6%	28,6%	1539	657	0	400	0	1057	0	0
<input checked="" type="checkbox"/>	546765877	Sněžná Anežka	75,0%	25,1%	25,0%	100,00	43,8%	95,2%	1507	2145	0	60	600	2805	0	0
<input type="checkbox"/>	567890432	Nová Eliška	75,0%	24,1%	25,0%	98,50	42,7%	47,6%	0	0	0	0	500	500	500	0
<input checked="" type="checkbox"/>	876545541	Antošová Eva	75,0%	21,4%	25,0%	100,00	41,1%	81,0%	0	0	0	3270	0	3270	0	0
<input type="checkbox"/>	393368999	Bulock Sandra	75,0%	23,9%	25,0%	92,00	40,9%	95,2%	0	0	0	0	0	0	0	0

Celkem: 7000 0 3730 600 11330 500 0

Zobrazit detail Obnovit/přepočítat Export operátorům Zadat ručně (osobní, tým, srážky) Zavřít

## Formulář F10: Evidence záznamů odpisů

**F10 Výkony a prémie 1.0 - GE Money CZ**

### Odpisové hodiny 2008

Uživatel: 678952478 Miroslav Hon Proces: DDTeam Dnešní datum: 8.3.2009

Aktuál..... Historie.....

Zadání nové odpisové hodiny

Položky čekající na schválení:

Datum	Typ odpisu	Čas	Poznámka
7.8.2008	Technický problém	35	Práce mimo systémy / Zajištění chodu procesu.

Upravit záznam Odstranit záznam

Neschválené položky:

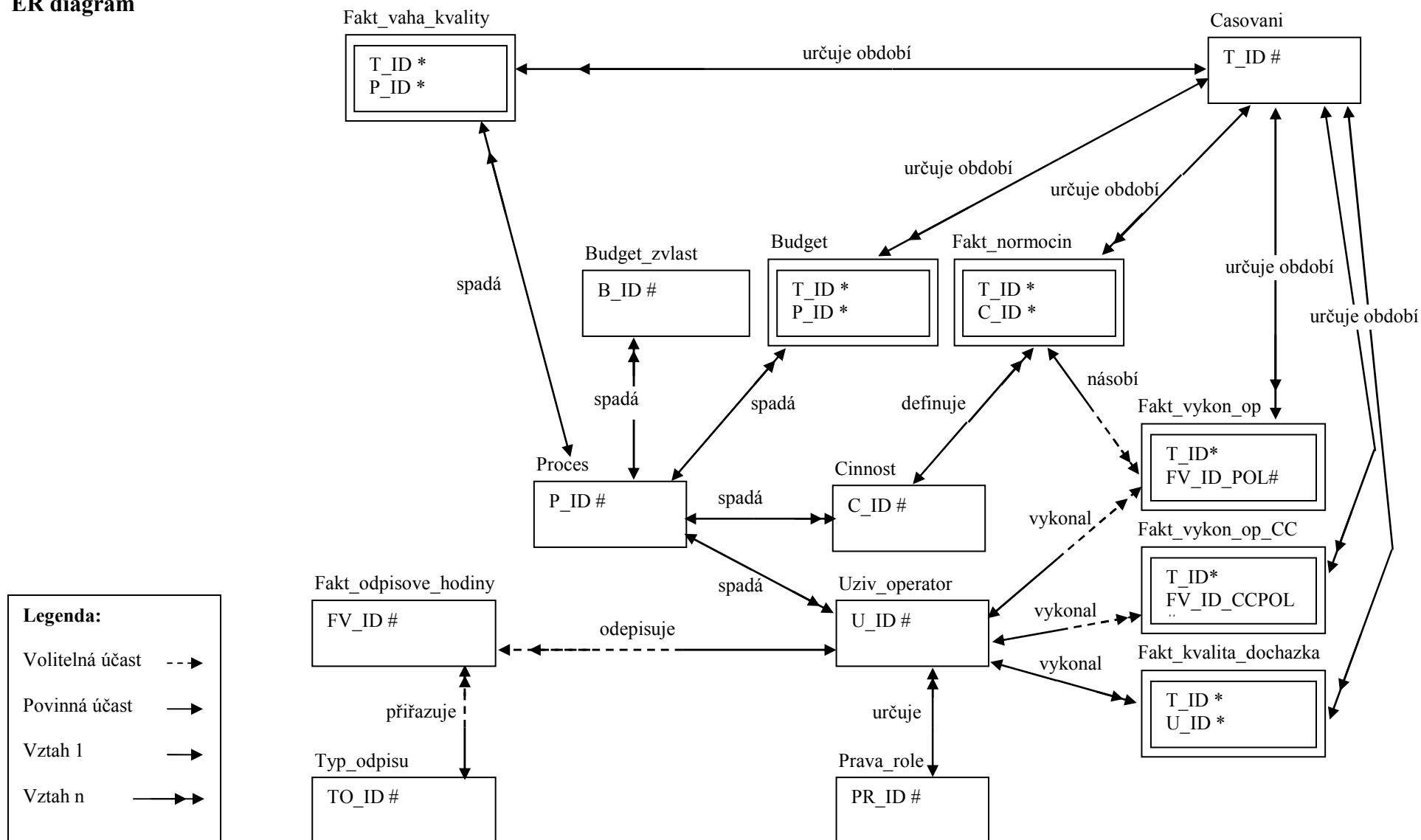
Datum	Typ odpisu	Čas	Poznámka
20.8.2008	Senior	1	
15.1.2009	Školení	5	

Upravit záznam Odstranit záznam

Zavřít

## Příloha č. 3 – Informační popis prototypu

### ER diagram



## Předběžné relace

Proces	(P_ID #, ...)
Uziv_operator	(U_ID #, ...)
Prava_role	(PR_ID #, ...)
Casovani	(T_ID #, ...)
Budget	(T_ID *, C_ID *, ...)
Budget_zvlast	(B_ID #, P_ID *, ...)
Cinnost	(C_ID #, ...)
Fakt_normocin	(T_ID *, C_ID *, ...)
Fakt_vaha_kvality	(T_ID *, P_ID *, ...)
Typ_odpisu	(TO_ID #, ...)
Fakt_odpisove_hodin	(FO_ID #, ...)
Fakt_vykon_op	(FV_ID_POL#, T_ID*, ...)
Fakt_vykon_op_CC	(FV_ID_CCPOL#, T_ID*, ...)
Fak_kvalita_dochazka	(T_ID*, U_ID *, ...)

## Úplné relace

Proces	(P_ID #, naz_proces)
Uziv_operator	(U_ID #, id_sso, jmeno, prijmeni, uvazek, kod_cc, kod_ufo, P_ID(C.K.), PR_ID(C.K.), )
Prava_role	(PR_ID #, popis)
Casovani	(T_ID #, od, do, duvod)
Budget	(T_ID *, P_ID *, castka, popis)
Budget_zvlast	(B_ID #, P_ID *, castka, popis)
Cinnost	(C_ID #, nazev, popis, normocas)
Fakt_normocin	(T_ID *, C_ID *, normocas)
Fakt_vaha_kvality	(T_ID *, P_ID *, vaha_kvality, vaha_kvantity)
Typ_odpisu	(TO_ID #, nazev, popis, P_ID(C.K.))
Fakt_odpisove_hodin	(FO_ID #, TO_ID(C.K.), U_ID(C.K.), datum, cas, poznamka, stav)
Fakt_vykon_op	(FV_ID_POL#, T_ID*, U_ID(C.K.), FN_ID(C.K.), pocet)
Fakt_vykon_op_CC	(FV_ID_CCPOL#, T_ID*, U_ID(C.K.), vykon)
Fakt_kvalita_dochazka	(T_ID*, U_ID *, pocet_hod, pocet_hod_mimo, kvalita, proc_plneni, tym_sloz, osobni_sloz, premie, sek, emu, poradí)

## Popis relací

**Proces** – číselník procesů.

Omezení	Atribut	Datový typ	Délka	Not null
PK	P_ID	Number		ANO
	naz_proces	Char	20	ANO

**Prava\_role** – číselník práv/rolí.

Omezení	Atribut	Datový typ	Délka	Not null
PK	PR_ID	Number		ANO
	popis	Char	200	ANO

**Uziv\_operator** – číselník uživatelů(operátorů) s osobními detaily.

Omezení	Atribut	Datový typ	Délka	Not null
PK	U_ID	Number		ANO
FK	P_ID	Number		ANO
FK	PR_ID	Number		ANO
UNIQUE	id_sso	Char	7	ANO
	jmeno	Char	20	NE
	prijmeni	Char	20	NE
	uvazek	Number		NE
UNIQUE	kod_cc	Char	6	ANO
UNIQUE	kod_ufo	Char	7	ANO

**Casovani** – číselník prémiových období.

Omezení	Atribut	Datový typ	Délka	Not null
PK	T_ID	Number		ANO
	od	Date		ANO
	do	Date		ANO
	duvod	Char	200	ANO

**Budget** – měsíční rozpočty jednotlivých týmů pro vyplacení prémie.

Omezení	Atribut	Datový typ	Délka	Not null
PFK	T_ID	Number		ANO
PFK	P_ID	Number		ANO
	castka	Number		ANO
	popis	Char	200	NE

**Budget\_zvlast** – oddělené rozpočty jednotlivých týmů (s platností na jeden kalendářní rok).

Omezení	Atribut	Datový typ	Délka	Not null
PK	B_ID	Number		ANO
PFK	P_ID	Number		ANO
	castka	Number		ANO
	popis	Char	200	NE



**Cinnost** – číselník všech činností vykonávaných operátory.

Omezení	Atribut	Datový typ	Délka	Not null
PK	C_ID	Number		ANO
UNIQUE	nazev	Char	100	ANO
	popis	Char	200	NE
	normocas	Number		ANO

**Fakt\_normocin** – data o přiřazeném normočasu jednotlivým činnostem na jednotlivá prémiová období.

Omezení	Atribut	Datový typ	Délka	Not null
PFK	T_ID	Number		ANO
PFK	C_ID	Number		ANO
	normocas	Number		ANO

**Fakt\_vaha\_kvality** – data o prémiových pravidlech (přiřazeny každému procesu a období).

Omezení	Atribut	Datový typ	Délka	Not null
PFK	T_ID	Number		ANO
PFK	P_ID	Number		ANO
	vaha_kvality	Number		ANO
	vaha_kvantita	Number		ANO

**Typ\_odpisu** – číselník typů odpisů. (odpis = činnost operátora, která doposud nelze systémově změřit).

Omezení	Atribut	Datový typ	Délka	Not null
PK	TO_ID	Number		ANO
FK	P_ID	Number		ANO
UNIQUE	nazev	Char	100	ANO
	popis	Char	200	NE

**Fakt\_odpisove\_hodin** – konkrétní odpisy operátorů. (operátor zadá činnost a počet hodin a tyto data jsou následně schválena nebo zamítnuta managerem týmu).

Omezení	Atribut	Datový typ	Délka	Not null
PK	FO_ID	Number		ANO
FK	TO_ID	Number		ANO
FK	U_ID	Number		ANO
	datum	Date		ANO
	cas	Number		ANO
	stav	Number		ANO
	poznámka	Char	200	NE

**Fakt\_vykon\_op** – informace o výkonech operátorů. Data jsou importovány a seskupeny z externích databází.

Omezení	Atribut	Datový typ	Délka	Not null
PK	FV_ID_POL	Number		ANO
PFK	T_ID	Number		ANO
FK	FN_ID	Number		ANO
FK	U_ID	Number		ANO
	pocet	Number		ANO

**Fakt\_vykon\_op\_CC** – informace o výkonech operátorů. Data jsou importována z externí databáze a liší se od předchozí relace strukturou dat.

Omezení	Atribut	Datový typ	Délka	Not null
PK	FV_ID_CCPOL	Number		ANO
PFK	T_ID	Number		ANO
FK	U_ID	Number		ANO
	vykon	Date		ANO

**Fakt\_kvalita\_dochazka** – informace o docházce operátorů, kvalitě odvedené práce a vyplacených prémiech, které se rozdělují do týmové složky, osobní složky, prémie za výkon a zvláštního ohodnocení (šek, EMU).

Omezení	Atribut	Datový typ	Délka	Not null
PFK	T_ID	Number		ANO
PFK	U_ID	Number		ANO
	pocet_hod	Number		ANO
	pocet_hod_mimo	Number		NE
	kvalita	Number		ANO
	proc_plneni	Number		NE
	tym_sloz	Number		NE
	osobni_sloz	Number		NE
	premie	Number		NE
	sek	Number		NE
	emu	Number		NE
	poradi	Number		NE

PK – primary key	FK – foreign key	PFK – primary foreign key
------------------	------------------	---------------------------